

Artificial Intelligence for Biology and Agriculture

Edited by

S. Panigrahi

North Dakota State University, Fargo, USA

and

K.C. Ting

Rutgers, The State University of New Jersey, USA

Reprinted from *Artificial Intelligence Review*
Volume 12, Nos. 1–3, 1998



SPRINGER SCIENCE+BUSINESS MEDIA, B.V.

VISIT...

LANZAROTE
Caliente.COM

A C.I.P. catalogue record for this book is available from the Library of Congress.

ISBN 978-94-010-6120-9 ISBN 978-94-011-5048-4 (eBook)
DOI 10.1007/978-94-011-5048-4

Printed on acid-free paper.

All Rights Reserved

©1998 Springer Science+Business Media Dordrecht

Originally published by Kluwer Academic Publishers in 1998

Softcover reprint of the hardcover 1st edition 1998

No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the copyright owner.

Table of Contents

About the Authors	1
Preface	9
M. MONTA, N. KONDO and K.C. TING / End-Effectors for Tomato Harvesting Robot	11
H. NI and S. GUNASEKARAN / A Computer Vision Method for Determining Length of Cheese Shreds	27
M. SLOOF / Automated Modelling of Physiological Processes During Postharvest Distribution of Agricultural Products	39
B.T. TIEN and G. VAN STRATEN / A Neuro-Fuzzy Approach to Identify Lettuce Growth and Greenhouse Climate	71
JONATHAN Y. CLARK and KEVIN WARWICK / Artificial Keys for Botanical Identification using a Multilayer Perceptron Neural Network (MLP)	95
MICHAEL RECCE, ALESSIO PLEBE, JOHN TAYLOR and GIUSEPPE TROPIANO / Video Grading of Oranges in Real-Time	117
ALAIN BOUCHER, ANNE DOISY, XAVIER RONOT and CATHERINE GARBAY / Cell Migration Analysis after <i>In Vitro</i> Wounding Injury with a Multi-Agent Approach	137
V.C. PATEL, R.W. McCLENDON AND J.W. GOODRUM / Color Computer Vision and Artificial Neural Networks for the Detection of Defects in Poultry Eggs	163
XIAOOU TANG, W. KENNETH STEWART, LUC VINCENT, HE HUANG, MARTY MARRA, SCOTT M. GALLAGER and CABELL S. DAVIS / Automatic Plankton Image Recognition	177
YOUNG J. HAN, YONG-JIN CHO, WADE E. LAMBERT and CHARLES K. BRAGG / Identification and Measurement of Convolutions in Cotton Fiber Using Image Analysis	201

BRIAN CENTER and BRAHM P. VERMA / Fuzzy Logic for Biological and Agricultural Systems	213
NAOSHI KONDO and K.C. TING / Robotics for Plant Production	227
K. DING and S. GUNASEKARAN / Three-Dimensional Image Reconstruction Procedure for Food Microstructure Evaluation	245

About the Authors

Alain Boucher graduated in computer engineering from the Ecole Polytechnique de Montréal, Canada, in 1994. He is currently doing his Ph.D. degree in computer science at the Université Joseph Fourier in Grenoble, France. For this degree, he received the 1967 Scholarship from the Natural Sciences and Engineering Research Council of Canada (NSERC). He works at the TIMC-IMAG laboratory, SIC group. His research interests include computer vision, distributed artificial intelligence and biomedical applications.

Charles K. Bragg, USDA-ARS, Cotton Quality Research Station, Clemson, SC 29631, USA

Brian Center was born in Decatur, Georgia in 1972. He graduated in Biological Engineering from Mississippi State University in 1994, and is currently working on the M.S. degree in Agricultural Engineering from the University of Georgia.

Since 1994 he has worked as a research assistant studying fuzzy logic applications related to plant growth models and plant tissue culture, and system identification and intelligent control of bioprocesses. He plans to attend medical school after completing his M.S. degree.

Yong-Jin Cho, Senior Research Scientist, Korea Food Research Institute, Songnam-si, Korea.

Jonathan Y. Clark is a Research Fellow in the Department of Cybernetics, Reading University, UK. He has an MSc

in Plant Taxonomy and has contributed to the Penguin Dictionary of Botany, and the European Garden Flora. He is a member of the British Cactus & Succulent Society, the Mesemb Study Group and the Systematics Association. His academic research has an emphasis on the use of computers and neural networks in biology, and in particular, botany. Although having a wide range of research interests, his main specialities are taxonomy and identification, especially with respect to cacti and other succulent plants, particularly the genus *Lithops* (which he has studied for nearly 25 years). In the field of cybernetics, his publications include the results of research into the application of neural networks for the identification of faults in high speed machinery.

Anne Doisy graduated in Pharmacy from the Faculté des Sciences Pharmaceutiques, Paris, in 1995. She is currently doing her Ph.D. degree in cellular and molecular biology at the Université Joseph Fourier in Grenoble, France. She works at Dyogen group, INSERM U 309, Institut Albert Bonniot in Grenoble. Her research is currently oriented towards the development of an in vitro model for cell migration analysis and is supported by the Ligue Nationale contre le Cancer.

Scott M. Gallager is an Associate Scientist in the Biology Department of the Woods Hole Oceanographic Institution. As a plankton ecologist, Scott seeks to understand the spatial and temporal scales of coupled biological and physical processes which control plankton

dynamics and community structure in the oceans. His long-term interest in optics and video electronics has led him to the co-development of the Video Plankton Recorder along with his colleague Dr. Cabell S. Davis. With this instrument, Scott is probing the mechanisms responsible for generation and maintenance of plankton swarms and the interaction between plankton behavior and turbulence. The goal is to use mechanistic models of physics and plankton behavior to assimilate data from the VPR in real-time and predict plankton dynamics and community structure.

Catherine Garbay graduated from Institut National Polytechnique de Grenoble (INPG) as a Computer Science Engineer (1977) and received her Ph.D. degree from INPG in 1979. She became "Docteur ès Sciences" from the Université Joseph Fourier and from the INPG, Grenoble in 1986. She is now employed by the CNRS as Research Associate and working at TIMC-IMAG laboratory, Grenoble. She was responsible for the creation, in 1988, of the SIC (Integrated Cognitive Systems) group, which is fostering research in the field of computer vision, distributed artificial intelligence and cognitive sciences, with application to biomedicine.

John W. Goodrum is a Professor in the Department of Biological and Agricultural Engineering at the University of Georgia. His research interests include applications of computer vision for the inspection and analysis of highly variable natural products.

Sundaram Gunasekaran is a professor of food and bioprocess engineering at the University of Wisconsin at Madison. He received his Ph.D. from University of Illinois in 1985 and worked as an assistant professor at the University of Delaware before moving to

Wisconsin in 1988. He received his B.E. (Ag.) in agricultural engineering from Tamil Nadu Agricultural University, Coimbatore, India and M. Eng. in food engineering from the Asian Institute of Technology in Bangkok, Thailand. Gunasekaran's research projects focus on enhancing quality of food materials via accurate and objective characterization of engineering properties. He is currently investigating rheology of dairy foods and mixed biopolymer gel systems. He is an active member of many professional societies including ASAE (American Society of Agricultural Engineers) and IFT (Institute of Food Technologists). He has been recognized for his research efforts with several national and international awards including the ASAE Outstanding Young Researcher Award in 1996 and ASAE Superior Paper Award in 1987 and 1995. He is the editor of Food and Process Engineering Institute for the Transactions of the ASAE and the Food Engineering Division of the IFT.

Young J. Han, Professor, Department of Agricultural and Biological Engineering, Clemson University, Clemson, SC 29634-0357, USA

He Huang received the B.S. degree in 1990 from the University of Science and Technology of China, Hefei, China, then received the M.S. degree in Mechanical Engineering, the M.S. degree in Ocean Engineering in 1994, and the Ocean Engineer degree in 1995, from the Massachusetts Institute of Technology. She is currently a design engineer at the Quantum Corporation. Her research interests include nonlinear dynamics system control, neural network control, and voice coil motor design in hard disk drives.

Naoshi Kondo was born in Ehime Prefecture, Japan in 1960. He received the BS, MS and PhD degrees in Agricul-

tural Engineering from Kyoto University, Japan in 1982, 1984 and 1988, respectively. From 1985 to 1990, he was with the Department of Agricultural Engineering of Okayama University as an Assistant Professor. From 1991 to 1992, he was with the graduate school of Natural Science and Technology of Okayama University as an Assistant Professor. Since April 1993, he has been an Associate Professor of Okayama University. His research interests are in the area of robotics for bio-production, machine vision and phytotechnology. Dr. Kondo is a member of the Japanese Society of Agricultural Machinery, the American Society of Agricultural Engineering, the Japanese Society of Environmental Control in Biology, the Society of Instrument and Control Engineers of Japan, the Robotics Society of Japan and so on.

Wade E. Lambert, Graduate Research Assistant, Department of Agricultural and Biological Engineering, Clemson University, Clemson, SC 29634-0357, USA

Marty Marra received the B.S. degree from Carnegie-Mellon University in Applied Math; Engineering Systems and Computer Science in 1985. He worked for Martin Marietta from 1985 to 1989 where he helped develop road following and terrain classification algorithms for DARPA's Autonomous Land Vehicle Program. In 1990 he joined the Deep Submergence Laboratory at the Woods Hole Oceanographic Institution where he developed systems for underwater photomosaicking, sonar mapping, and video plankton detection. Since 1995 he's worked for Vexcel Corporation where he's managed projects applying synthetic aperture radar data to stereo topographic mapping and terrain classification. His research interests involve the development of imaging, visualization, and com-

puter vision systems for remote sensing and mapping applications.

Ronald W. McClendon is a Faculty Fellow of the Artificial Intelligence Center and Professor in the Department of Biological and Agricultural Engineering at the University of Georgia. His research interests involve the application of AI and operations research techniques to the development of decision support systems.

Mitsuji Monta was born in Osaka Prefecture, Japan in 1961. He received the BE and ME degrees in Agricultural Engineering from Okayama University, Japan in 1986, 1988, respectively. From 1988 to 1991, he was with the Kubota Corporation in Osaka, Japan. Since June 1991, he has been an Assistant Professor of Department of Agricultural Engineering of Okayama University. His research interests are multi-purpose robots for bio-production, safety robot systems and phytotechnology. He is a member of the Japanese Society of Agricultural Machinery, the American Society of Agricultural Engineering, the Japanese Society of Environmental Control in Biology, the Robotics Society of Japan and so on.

Hongxu Ni is a Ph.D. candidate in the Biological Systems Engineering Department at the University of Wisconsin-Madison. Mr. Ni earned his MS degree in Computer Sciences from University of Wisconsin-Madison in 1995 and M. Eng. in Computer Engineering from Southwest Jiaotong University in China in 1992. Since 1992, Hongxu has been conducting research and development in computer vision, visual simulation, and real-time system programming. He is currently working as a Software Engineer at Teradyne Inc., Los Angeles, CA.

Suranjan Panigrahi (guest editor), is an Assistant Professor of Agricultural and Biosystems Engineering Department, North Dakota State University, Fargo, ND. He obtained his Ph.D. in Agricultural and Biosystems Engineering with a Ph.D. minor in Electrical & Computer Engineering from Iowa State University. His research and teaching interests are in the areas of artificial intelligent technologies (computer vision, pattern recognition, neural networks, fuzzy logic) for agricultural/food production & processing applications, non-visible imaging/spectrometry, non-destructive and intelligent sensor/sensing techniques for high quality and safe food production and processing. He is a honorary member of Alpha Epsilon and Gamma Sigma Delta.

Viren C. Patel is a post doctoral fellow at the University of Texas – Houston Medical School and was formerly a graduate research assistant in the Department of Biological and Agricultural Engineering at the University of Georgia. His research interests involve the application of engineering and artificial intelligence techniques to the solution of problems in biological systems.

Alessio Plebe received a diploma in Electronic Engineering in 1981 at the University of Rome. He worked at AID (Agriculture Industrial Development) SpA, Catania, Italy from 1982 to 1989 as researcher in Applied Meteorology, where he developed models for short-time forecasting and a system for local-scale artificial weather modifications. From 1990 to 1996 he led the Robotics and Automation Department at C.R.A.M. (Consorzio per la Ricerca in Agricoltura nel Mezzogiorno), Catania, Italy. During this period he studied applications of robotics and visual inspection in the field of agriculture and food processing. He lead several European

Projects for AID and CRAM in the fields of image processing, robotics and neural networks. His current area of research also includes educational technologies. He is member of the Scientific Board of EUROLOGO, and he is temporary professor of Computer Science at the Educational Science Faculty, University of Palermo, Italy.

Michael Recce received a BSc in Physics from the University of California at Santa Cruz. He joined Intel Corporation in 1982 where he participated in the development of magnetic bubble memory products, and later led the product engineering group for bubble memories. He obtained a PhD in neurophysiology at University College London (UCL), and in 1991 became a joint lecturer in the Anatomy and Computer Science Departments at UCL. At UCL he led research projects in neurophysiology, computational neuroscience and robotics. In 1997 his research group moved to the New Jersey Institute of Technology, where he is an Associate Professor in the Computer Science and Information Science and the Applied Mathematics Departments. He is also a member of the Behavioral Neuroscience Program at Rutgers University.

Xavier Ronot received his Doctorat ès Science in Biology from Université Paris VII in 1987. He is currently Assistant Professeur at the Ecole Pratique des Hautes Etudes and is working at group Dyogen, INSERM U 309, Institut Albert Bonniot, at the Université Joseph Fourier in Grenoble. His main research interests concern the designing of in vitro models to study the structure-function relationships in living cells, especially oriented towards cell proliferation, chromatin organization and migration using vital fluorescent probes.

Mark Sloof received his MS in computer science from the University of

Leiden in 1991. In that year he joined the Artificial Intelligence Group at the Vrije Universiteit in Amsterdam to work on a research project initiated by and carried out at the Agricultural Research Institute (ATO-DLO) in Wageningen. His research interests are the application of qualitative reasoning techniques to support the development of quantitative models. He is currently employed as knowledge engineer with Everest in 's-Hertogenbosch, The Netherlands, which specializes in applying techniques for software reuse to the development of practical AI systems.

W. Kenneth Stewart received the A.A.S. in Marine technology from Cape Fear Technical Institute in 1972, the B.S. in Ocean Engineering from Florida Atlantic University in 1982, and the Ph.D. in Oceanographic Engineering from the Massachusetts Institute of Technology and Woods Hole Oceanographic Institute Joint Program in 1988. Stewart has been going to sea on oceanographic research vessels for more than 20 years, has developed acoustic sensors and remotely operated vehicles for 6000-m depths, and has made several deep dives in manned submersibles, including a 4000-m excursion to the Titanic in 1986. Since 1988 he has been a member of the scientific staff at the Woods Hole Oceanographic Institution and is now an Associate Scientist at the Deep Submergence Laboratory. His research interests include underwater robotics, autonomous vehicles and smart ROVs, multisensor modeling, real-time acoustic and optical imaging, and precision underwater surveying. He is a member of the Acoustic Society of America, IEEE Computer Society, Marine Technology Society, National Computer Graphics Association, Oceanography Society, and Sigma Xi.

Gerrit van Straten (1946) previously held positions in environmental engineering at Twente University, the Netherlands, and the International Institute for Applied Systems Analysis (IIASA), in Laxenburg, Austria. He received a Ph.D. degree at Twente University on 'Identification, uncertainty assessment and prediction in lake eutrophication'. Since 1990 he holds a chair in Systems and Control at Wageningen Agricultural University. His group's task is to advocate the use of systems modelling and control in a broad spectrum of applications in agriculture, food processing and environmental technology. Current research interests are modelling, identification and uncertainty, optimal control and predictive control, and neural and fuzzy controllers, with applications to horticulture (greenhouse control), sewage treatment, water systems, food processing and bio-engineering.

Xiaoou Tang received the B.S. degree in 1990 from the University of Science and Technology of China, Hefei, China, and the M.S. degree in 1991 from the University of Rochester, Rochester, New York. He received the Ph.D. degree in 1996 from the Massachusetts Institute of Technology in the MIT/Woods Hole Oceanographic Institution Joint Program. He is currently a postdoctoral investigator at the Deep Submergence Laboratory of Woods Hole Oceanographic Institution. His research interests include image processing, pattern recognition, underwater robotics, and sonar systems.

John C. Taylor obtained a B.A. (hons) degree in Physics from Fitzwilliam College at Cambridge University and a M.Sc. degree in Systems Engineering from the City University in London. For the past twenty years he has worked in research and development in the areas of software engineering and pattern recognition algorithms. He is currently employed

by Searchspace Limited in London as a senior software engineer, where he is developing reporting and analysis systems for the London Stock Exchange. From 1988 to 1997 he was a senior research fellow at University College London. At UCL he designed and developed software tools for real-time neural network based pattern recognition.

Biing-Tsair Tien received the BS degree in Agricultural Machinery Engineering, National Taiwan University, Taipei, Taiwan, Republic of China, in 1987, the MSc degree from the Institute of Agricultural Engineering, National Taiwan University in 1989 and the PhD degree in the Department of Agricultural, Environmental and Systems Technology, Wageningen Agricultural University, The Netherlands, in 1997. Based on his MSc researches he won the Best Paper Award from the Society of Chinese Agricultural Engineering in 1990. During 1991–1993, he had been with the National Taiwan University as an instructor in the department of Agricultural Machinery Engineering. Presently, he works with the Tokyo Electron Taiwan Limited Co., as a MCVD (Metal Chemical Vapor Deposition) process engineer. His research interests include Neural-Fuzzy modeling and the application of Mechatronics technology to agricultural automation issues.

K. C. Ting, is a Professor and Chairman of Bioresource Engineering Department, Rutgers – The State University of New Jersey, received his Ph.D. in agricultural engineering from the University of Illinois. He is currently the Editor of ASAE/Information and Electrical Technologies Division. He is a licensed professional engineer in New Jersey. His research and teaching interests are in the areas of flexible automation and robotics for bio-processing and bio-production, systems analysis and decision support engineering, phytomation

engineering, and advanced life support systems for long term human exploration of space.

Giuseppe Tropiano received a diploma in Computer Science from the University of Pisa in 1993. From 1995 to 1996 he worked at AID (Agriculture Industrial Development) Catania, Italy, doing research in image processing and neural networks applied to fresh fruit inspection. He is currently employed by the National Pole of Bioelectronics (Elba, Italy), where he is applying neural network algorithms to visual inspection tasks.

Brahm Verma is Professor of Biological and Agricultural Engineering and Faculty Fellow in the Artificial Intelligence Center at the University of Georgia. He received his B.Sc. from the University of Allahabad in India, M.S. from the University of Kentucky and Ph.D. from Auburn University. He has conducted research in similitude, tillage machinery, nursery and greenhouse mechanization, postharvest systems and simulation and modeling. He is currently leading a research center for developing decision support system tools to evaluate economic development opportunities that are environmentally sustainable. He also serves as the Graduate Coordinator of the department.

Luc Vincent is Director of Software Development at Xerox' Software Solutions Division in Palo Alto, CA, where he heads the Image Processing Core Technology group. He received the Engineering Degree from Ecole Polytechnique, France, in 1986, and the Doctorate in Mathematical Morphology from the Ecole des Mines de Paris in 1990. Dr. Vincent then worked at Harvard University as a Postdoctoral Fellow, and joined Xerox in 1991, where he has worked on various aspects of OCR and

document image analysis, with emphasis on preprocessing, compression, and segmentation issues. Over the past ten years, Luc Vincent has been involved in a variety of image processing projects, in such areas as medical applications, biology, industrial inspection, oil exploration, oceanography, etc. His main interests are in image segmentation, feature extraction, and algorithm design, and he has published over fifty papers on such topics. He is currently an associate editor of the *Journal of Electronic Imaging*, and has served as chairman for over a dozen conferences since 1991. His book on morphological image analysis will be published by Cambridge University Press in 1998.

Kevin Warwick is Professor of Cybernetics at the University of Reading, UK. He previously held positions at Oxford, Newcastle and Warwick Universities and has higher Doctorates from both Imperial College, London and the Czech Academy of Sciences, Prague. He is on the Editorial Board of several international journals and is an Honorary Editor of the *IEE Proceedings on Control Theory and Applications*. Kevin's research interests lie mainly in machine intelligence, computer control and robotics and he has published over 300 articles including 21 books on these topics. His main thrust at present is in the study and application of novel machine intelligence methods across a range of implementations.

Preface

Biological and agricultural discipline and their associated environment, processes and systems/subsystems are very crucial for a better, sustainable and quality living of human beings. Thus, continuous investigations and studies have been undertaken in the past, are going on in the present and will be continued in the future to understand, explore and solve different problems associated with various systems/subsystems of biological and agricultural discipline. However, inherently complex, dynamic and non-linear characteristics of the biological/agricultural systems have always required solutions based on advanced techniques and technologies to provide higher accuracies, better understandings and appropriate solutions.

In recent years, the advent of artificial intelligent technology (i.e. computer vision, robotics and control systems, expert systems/decision support systems, natural language processing, etc.) and other advanced forms of information technology (neural networks, fuzzy logic and genetic algorithms) have shown promises for finding solutions to different biological/agricultural systems. The advancements of the technologies with their decreasing cost is catalyzing additional investigations on the applications of different forms of AI technologies in these disciplines. Therefore, this special issue “**Artificial Intelligence in Biology and Agriculture**” was developed to report on a sampling of selected (peer-reviewed) state-of-the art research directed towards solving different problems in the biological and agricultural discipline.

This issue contains a total of thirteen papers covering a variety of AI topics ranging from computer vision and robotics to intelligent modeling, neural networks and fuzzy logic. There are two general articles on robotics and fuzzy logic. The article on robotics focuses on the application of robotics technology in plant production. The second article on fuzzy logic provides a general overview of the basics of fuzzy logic and a typical agricultural application of fuzzy logic. The article “*End effectors for tomato harvesting*” enhances further the robotic research as applied to tomato harvesting. The application of computer vision techniques for different biological/agricultural applications i.e, length determination of cheese threads, recognition of plankton images

and morphological identification of cotton fibers depicts the complexity and heterogeneities of the problems and their solutions. The development of a real-time orange grading systems in the article "*Video grading of oranges in real-time*" further reports the capability of computer vision technology to meet the demand of high quality food products. The integration of neural network technology with computer vision and fuzzy logic for defect-detection in eggs and identification of lettuce growth shows the power of hybridization of AI technologies to solve agricultural problems. Additional papers also focus on automated modeling of physiological processes during postharvest distribution of agricultural products, the applications of neural networks, fusion of AI technologies and three dimensional computer vision technologies for different problems ranging from botanical identification, cell migration analysis to food microstructure evaluation.

This special issue "**Artificial Intelligence in Biology and Agriculture**" has been made possible due to the unconditional help, cooperation and time devotion from many people. We highly appreciate the contributions from the authors and their co-authors. We sincerely acknowledge all reviewers for taking time to review these articles. The reviewers were: Dr. Kuanglin Chao, Dr. Floyd E. Dowell, Dr. Laurent Gauthier, Dr. Paul H. Heinemann, Dr. Zhiwei Li, Dr. Bosoon Park, Dr. Jinglu Tan, Dr. Chi Ngoc Thai, Dr. Basant Ubhaya, Dr. Naiqian Zhang, Dr. Irfan Ahmad, Dr. David Vacaari, Dr. Young Han, Dr. Lary Kutz, Dr. David Slaughter, Dr. Digvir Jayas, Dr. Marvin Paulsen, Dr. George Hoogenboom, Dr. Mark Evans, Dr. Glen Kranzler, and Dr. Jim Lindley. We express our thanks to the US editor Dr. Evangelos Simoudis for his guidance and providing us with this opportunity. We are also very thankful to all the editorial staff of Kluwer Academic publishers for their cooperation and time to make this special issue a reality.

Sincerely,
The Guest Editors

Suranjan Panigrahi
K.C. Ting

End-Effectors for Tomato Harvesting Robot

M. MONTA¹, N. KONDO¹ and K.C. TING²

¹*Agricultural Engineering Dept., Faculty of Agriculture, Okayama University, 1-1-1, Tsushima-Naka, Okayama, Japan;* ²*Dept. of Bioresource Engineering, Rutgers University-Cook College, P.O. Box 231, New Brunswick, New Jersey 08903-0231, USA*
(E-mail: monta@ccews2.cc.okayama-u.ac.jp)

Abstract. Two types of robotic end-effectors capable of harvesting tomato fruits were manufactured based on the physical properties of tomato plant and tested. The first prototype end-effector consisted of two parallel plate fingers and a suction pad. The fingers pick a fruit off at the joint of its peduncle after the suction cup singulates it by vacuum from other fruits in the same cluster. From the results of harvesting experiment, the end-effector could not harvest fruits with a short peduncle because the fruits were detached from the suction pad before they were gripped by the fingers. Therefore, the second prototype in which the functions to detect the fruit position and the air pressure in the pad were installed, was made, so that the fruits were harvested regardless of the length of their peduncle. Experimental results using the improved end-effector showed that the fruits were harvested successfully with no damage.

Key words: robotics, tomatoes, harvesting, end-effector, manipulator, sensors

1. Introduction

Robotic end-effectors for agricultural operations such as harvesting, spraying, transplanting and berry thinning have been developed in recent years [1, 2, 3, 4, 5]. Robotic end-effectors are important components in the development of agricultural robots, because they handle plant materials directly and can potentially influence the market value of the product. As for tomato, a two-plate hand, a hand which harvests fruits by guided rotation and other types of end-effectors have been studied for harvesting [6, 7]. However, they often injure neighboring fruits or stems during harvesting.

In this study, two types of robotic end-effectors were manufactured based on the physical properties of tomato plant, to harvest fruits with no damage. The first prototype end-effector consisted of two parallel plate fingers and a suction pad. The fingers pick a fruit off at the joint of its peduncle after the suction cup singulates it by vacuum from other fruits in the same cluster. Harvesting experiment was carried out after the end-effector was attached to a manipulator with 7 degrees of freedom. From the result, the end-effector could not harvest fruits with a short peduncle. Then, the second prototype in which improvements were made on the first prototype was manufactured and

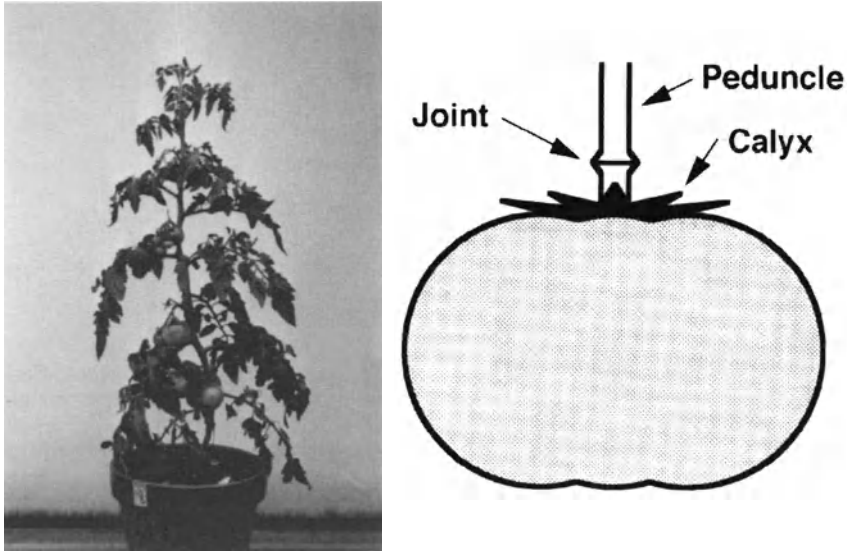


Figure 1. Tomato plant grown vertically and tomato fruit.

tested. The development of the tomato harvesting end-effectors and the test results are described in this article.

2. Cultivation Method of Tomato Plant

The most common training method of tomato (*Lycopersicon esculentum* M.) plant for flesh market in Japan is to use vertical support so that the plant grows upward as shown in Figure 1. The plant is transplanted so that the clusters of fruit direct to aisle side of ridge, and continues setting a series of clusters of fruit until the top growing tip is pruned out. The cluster has several fruits which are adjacent to one another. In many varieties of the tomatoes, on the peduncle of a tomato fruit there is a joint which is easy to be served by bending. Therefore not only human worker but also robot can pick off fruit one by one easily by bending instead of cutting.

3. Harvesting Experiment Using First Prototype End-Effector

3.1. *Experimental apparatus*

Figure 2 shows a manipulator used in this experiment. The basic mechanism of manipulator which was adapted to the physical properties of tomato plant was investigated by use of evaluating indexes such as operational space, measure of manipulability, space for obstacle avoidance, redundant space and posture diversity [8]. The manipulator which was made based on the result of evaluation had 7 degrees of freedom including 2 prismatic joints (s_1 and s_2) and 5 rotational joints ($\theta_3, \theta_4, \theta_5, \theta_6$ and θ_7). The prismatic joints were used mainly for overall manipulator positioning and the rotational joints were mainly for manipulator posture [9]. The lengths of upper arm (l_4) and fore arm (l_5) were 250 mm and 200 mm, respectively, while the strokes of the prismatic joints were 200 mm (horizontal direction) and 300 mm (vertical direction), respectively.

Figure 3 shows a first prototype end-effector [10]. It consists of two parallel plate fingers and a suction pad between the fingers. The fingers pick a fruit off at the joint of its peduncle after the suction pad singulates it by vacuum from other fruits in the same cluster. A 10 mm thick rubber pad is attached to each finger plate to prevent fruit from slipping and damage. The length, width and thickness of a finger plate are 155 mm, 45 mm and 10 mm, respectively. The gripping force exerted by the finger plates can be adjusted from 0 to 33.3 N, while these finger plates grip fruits ranging from 50 to 90 mm in diameter.

A suction pad is suitable for pulling, holding and twisting a fruit when a robot handles the delicate plants [11, 12], and has several advantages that it can fit to the shape of object and compensate a certain degree of positioning error by a fruit location sensor. The suction pad made of silicon is attached to the end of a rack which is driven back and forth by a DC motor and a pinion in between the finger plates. The cross-sectional area and the capacity of suction pad are 1.84 cm^2 and 1.5 cm^3 respectively. The sucking force which is supplied by a vacuum pump through a solenoid valve is about 10 N. The solenoid valve changes air flow direction to effectively create suction force and release fruit. The pad can be moved forward up to 15 mm from the tips of the finger plates. A touch sensor attached to the suction pad turns on when the suction pad sucks a fruit. Two limit switches attached to both ends of the pad stroke detect the stop positions of the suction pad. A parallel interface is used between a computer (8086 cpu) and the peripheral devices. The rotating direction of the DC motor and switching of the vacuum pump and the solenoid valve are controlled by switching electro magnetic relays.

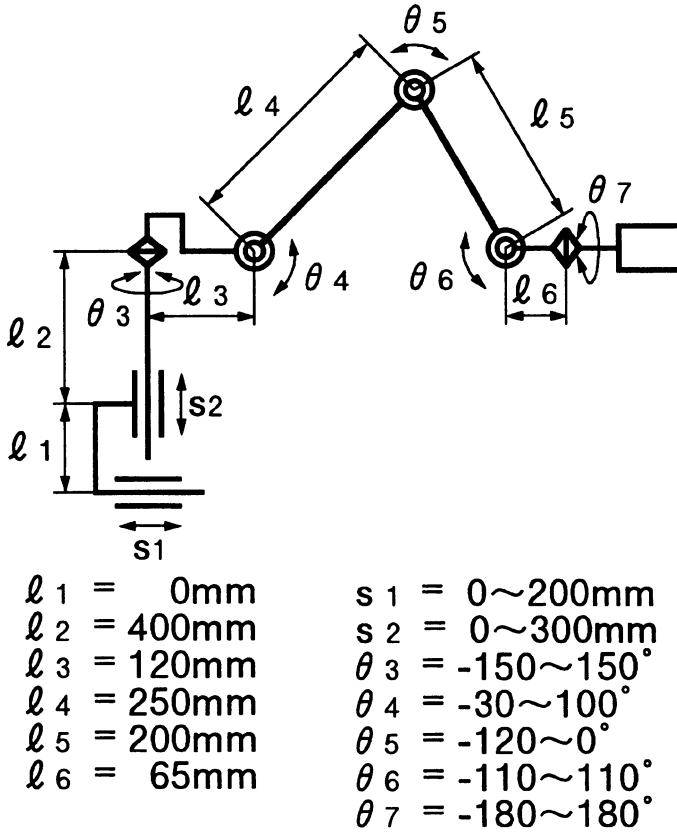


Figure 2. Construction of manipulator.

3.2. Experimental method

The tomato harvesting experiment was carried out by using the manipulator and the end-effector. After the robot assumed an initial posture perpendicular to the tomato row, the three-dimensional position of a target fruit and the wrist rotational angle corresponding to a inclination of the fruit were entered into the computer manually since no fruit location sensor was used in this experiment. If the objective fruit existed within the working area of manipulator, the manipulator moved toward the fruit. In this experiment, two approaching methods of manipulator were adopted; one method that the end-effector approached a fruit horizontally, and the other method that the end-effector approached a fruit at an angle of 45 degree under with respect to the horizontal if obstacles existed in front of the target fruit. At an appropriate position, the

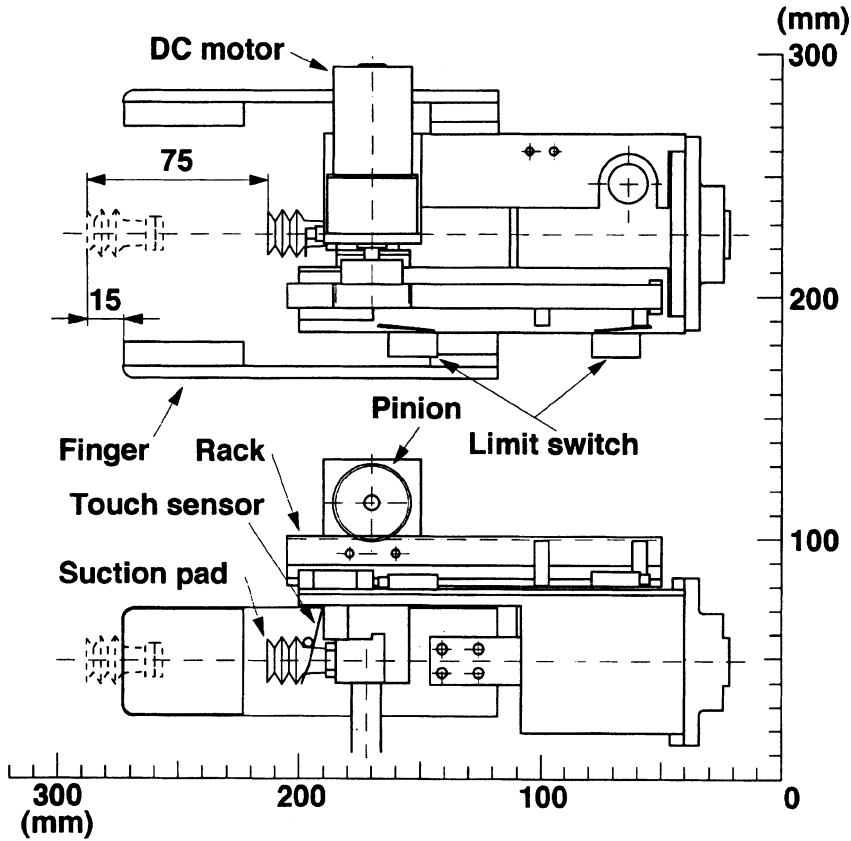


Figure 3. End-effector (First prototype).

suction pad moved toward the fruit after the wrist joint was rotated and the vacuum pump and the solenoid valve were turned on. As soon as the touch sensor was turned on, the suction pad moved backward to singulate the target fruit from the others in the same cluster. If the suction pad could not suck the fruit, the end-effector moved forward more 10 mm. Finally, the finger plates gripped the fruit and the end-effector harvested the fruit by bending at the peduncle and released the fruit in a tray.

3.3. Result and discussion

Figure 4 shows a example of the motions of manipulator when the end-effector approached the target fruit at an angle of 45 degree under with respect to the horizontal. The manipulator assumed an initial posture after moved upward

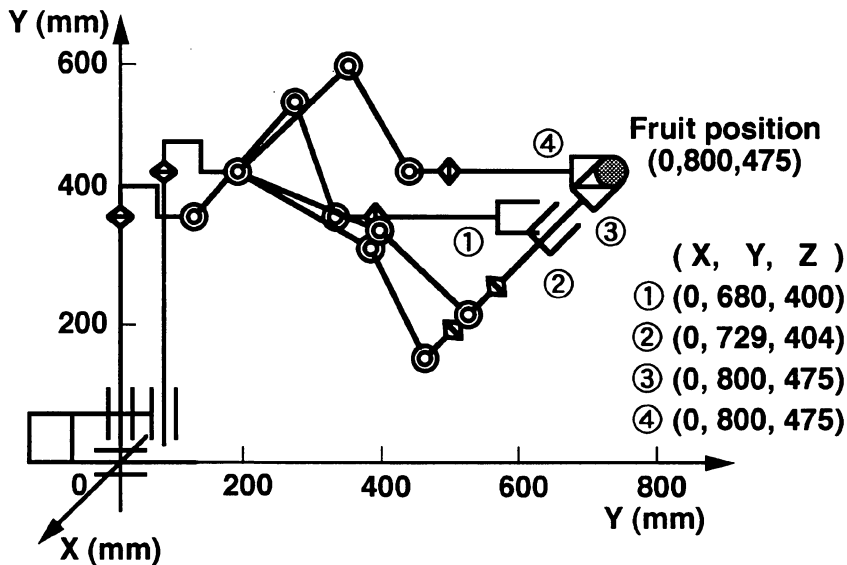


Figure 4. Motions of manipulator for fruit harvesting.

by the vertical prismatic joint motion, furthermore moved forward by the horizontal prismatic joint so that the fore arm pointed to the fruit. Then the end-effector was moved straight toward to the fruit until the touch sensor was turned on. Finally, the fruit was gripped by fingers after being separated from others, and harvested.

Tomato fruits (cv. "Saturn") including mature and over-mature fruits were used for the harvesting experiment. From the result of experiment, about 85% of fruits were harvested by the robot. The largest cause for the fruits which were not harvested was that they detached from the suction pad before they were gripped by the fingers. Table 1 shows the average peduncle length and peduncle diameter of the harvested fruits, and those of the fruits separated from the pad. It was observed that the average peduncle length of the detached fruits (average of over-mature and mature fruits: 37.5 mm) was shorter than that of the harvested fruits (52.6 mm), though there was no obvious difference between the harvested and detached fruits in their average peduncle diameter (5.1 mm and 5.2 mm respectively). Therefore, it was considered that the fruits with a short peduncle detached from the suction pad due to the short fruit moving distance away from its original position. In addition, some fruits which were partially hidden behind other fruit could not be harvested because the suction pad could not reach them.

Table 1. Average peduncle length and preduncle diameter of fruit

	Peduncle length (mm)		Peduncle diameter (mm)	
	Over-mature	Mature	Over-mature	Mature
Harvested (45)	49.0 (30)	54.9 (15)	5.1 (30)	5.0 (15)
Detached (8)	40.3 (6)	29.0 (2)	4.8 (6)	6.5 (2)

Number within () indicates quantity of fruit.

Therefore, it was considered that the end-effector required the capabilities to detect the change of vacuum pressure and moving distance of the pad, and to reach the farther fruits. In order to solve these problems, some functions of the end-effector were improved and the harvesting experiment was carried out.

4. Harvesting Experiment Using Second Prototype End-Effector

4.1. *Experimental apparatus*

Figure 5 shows the improved end-effector which mounts a pressure sensor and a potentiometer to monitor the pressure and position of the pad respectively. Furthermore, a stroke of the pad is extended 28 mm longer than that of the first prototype, consequently the pad can be moved forward up to 43 mm from the tips of the finger plates. The moving distance and stopping position of the pad can be detected by a rotary type potentiometer. Two limit switches are attached to both ends of the pad stroke in order to prevent the pad from overrunning. Figure 6 shows the end-effector attached to the manipulator.

Figure 7 shows a check valve connected to the suction pad. A pressure sensor which is connected to the check valve through a tube measures air pressure in the suction pad. The analog voltage (0 to 5v) output from this sensor is proportional to the gage pressure which ranges from -0.1 to 0 MPa. The size of pressure sensor is 46.5 mm long, 28 mm wide and 16mm thick and weighs 20 g. The check valve between the suction pad and the vacuum pump regulates the direction of air flow to measure the air pressure change without causing the expansion and contraction of the tube. The air flows from the pad to the vacuum pump through the valve when the pad does not have a fruit attached to it. If the pad attaches to a fruit and retrieves the fruit toward the hand, the valve closes automatically and the air flow is stopped, since the air pressure in the pad becomes lower than that of the pump side due to the expansion of the pad. This valve also minimizes measurement errors of air pressure change caused by leakage through the pump. The cracking

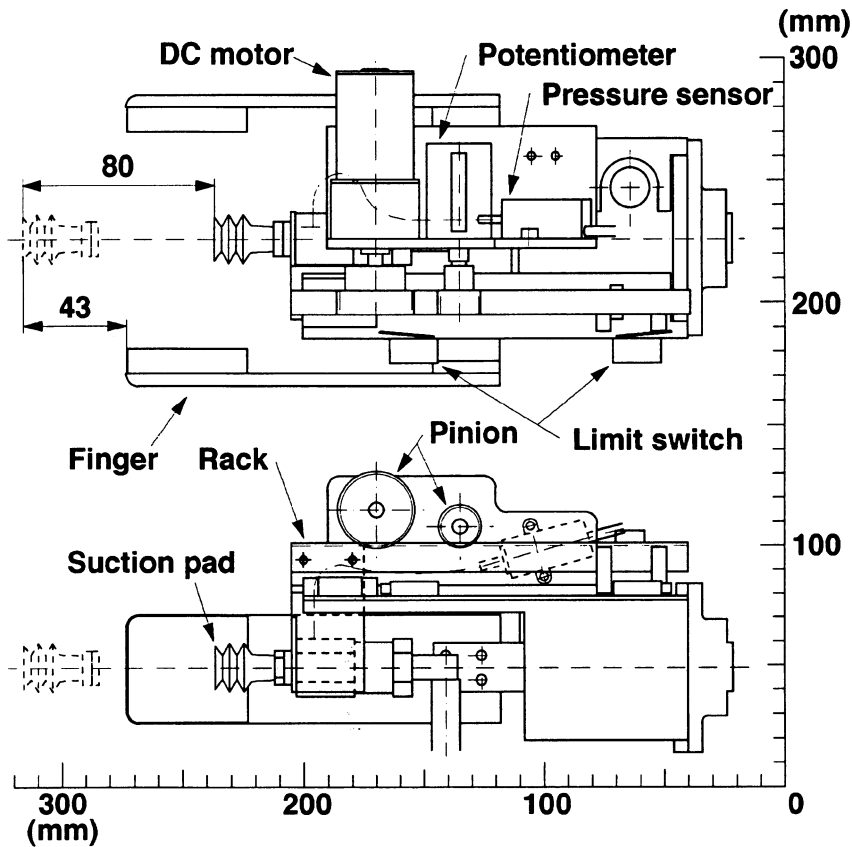


Figure 5. End-effector (Second prototype).

pressure of the check valve is 0.015 MPa and the effective cross-sectional area is 0.1 cm^2 . The volumetric capacity including pad, tube and check valve is approximately 4.5 cm^3 when the valve is closed and 3.5 cm^3 when the pad is fully contracted.

The output voltages from the pressure sensor and potentiometer are amplified and converted to digital signals by an 8-bit A/D converter, so that the pressure in the pad and the position of the pad can be detected. The pressure and moving distance are measured at 50 ms intervals within 1 kPa and 1 mm accuracy, respectively. The limit switches which are connected between the DC motor and the power supply, shut off the circuit when the pad moves beyond the stop positions.

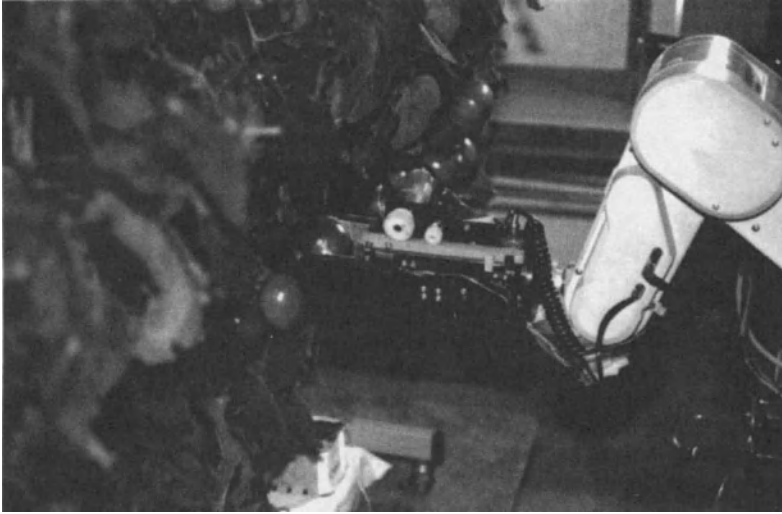


Figure 6. End-effector attached to manipulator.

4.2. *Experimental method*

4.2.1. *Control method*

The tomato harvesting experimental procedure was carried out after the improved end-effector was attached to the manipulator end. After the vacuum pump and the solenoid valve were turned on and the robot assumed an initial posture, the three dimensional position of a target fruit was entered into the computer manually since no fruit location sensor was used in this experiment. The manipulator moved the end-effector toward the target fruit. At an appropriate end-effector position, the suction pad moved toward the fruit while its pressure and position were constantly monitored. As soon as the pressure became equal to a predetermined value, the pad stopped and started to move backward after the vacuum pump was turned off. This pad motion was designed to singulate the target fruit from the others in the same cluster. After the pressure in the pad reached another set value, the entire end-effector started to move toward the fruit at the same speed as the suction pad simultaneously. This enabled the fruit to stay at a constant absolute position. Finally, the finger plates gripped the fruit and the end-effector harvested the fruit by bending at the peduncle and released the fruit in a tray (Figure 8).

4.2.2. *Determination of set values*

Figure 9 shows the relationship between pad movement measured by potentiometer and change of pressure in the pad detected by the pressure sensor.

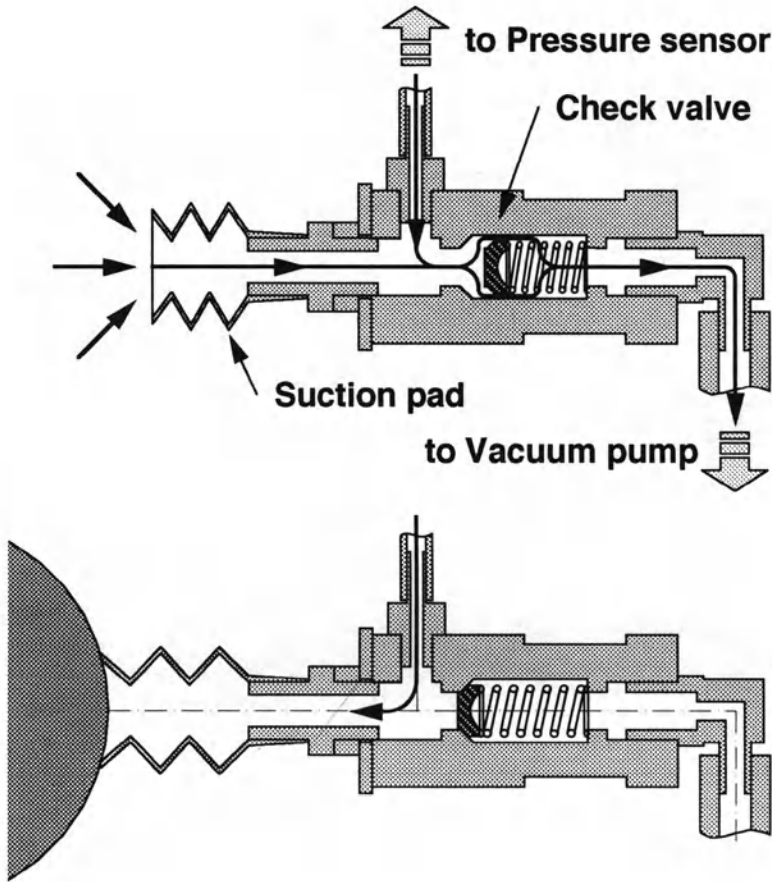
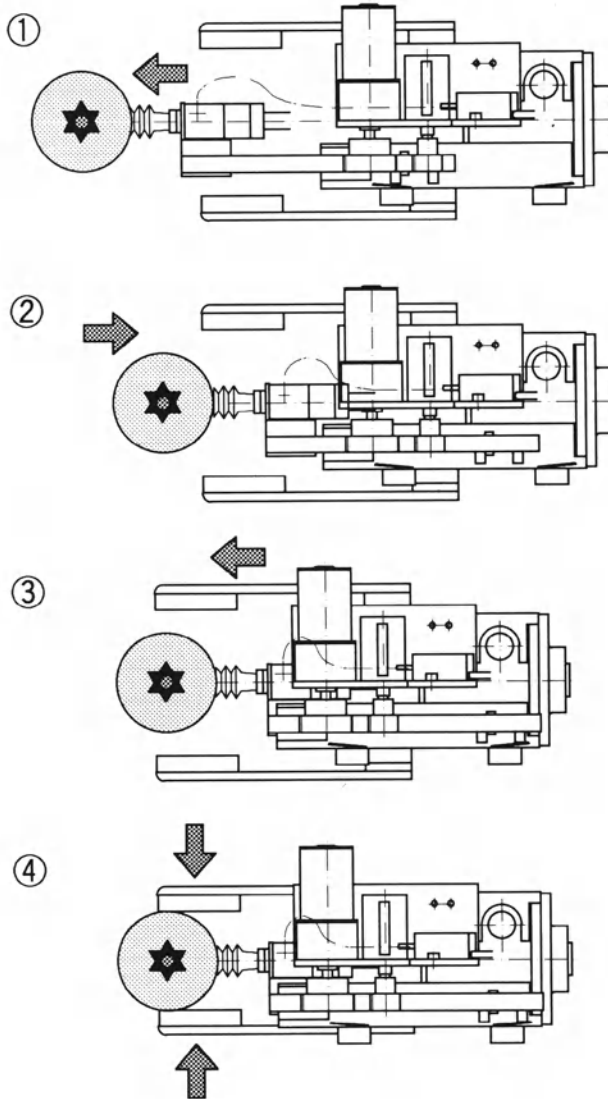


Figure 7. Motion of suction pad and check valve.

The vertical axis indicates pressure and moving distance and the horizontal axis indicates elapsed time. This figure is a result of the condition under which the fruit was picked by the suction pad and moved backward at a speed of 37.7 mm/s but the end-effector did not move forward. This procedure was performed to determine the negative pressure when a fruit separated from the suction pad. The set value 1 of pressure was determined as -20 kPa, when the fruit was firmly attached to the pad based on the preliminary experiment. In the harvesting experiment, the pad was controlled to stop moving forward when the pressure reached the set value 1, and to attach a fruit for 1 s. The pressure continued to decrease gradually and reached eventually the stable pressure value P , in this case it was around -33 kPa. When the pad started



moving backward, the pressure continued to decrease. Finally, the fruit separated from the pad at -44 kPa when the pad could no longer hold the fruit. This indicated that the fruit was separated from the pad when the negative pressure reached 1.3 times the stable pressure value P . Therefore, the set value

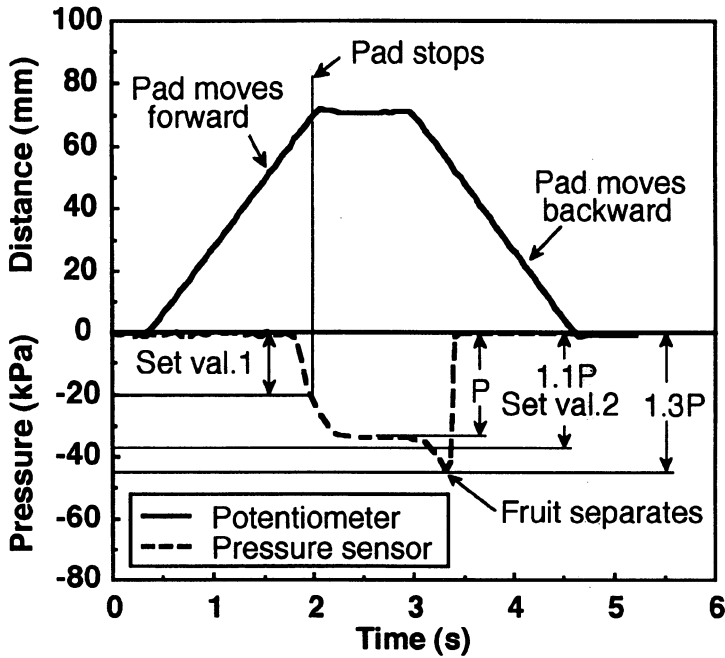


Figure 9. Relation between moving distance and pressure for determining set values.

2 for the pad to stop pulling a fruit was set to be 1.1 times the pressure value P in order to prevent a fruit from leaving the pad.

4.3. Results and discussion

Figure 10 shows an example of experimental results. It was observed that the pad stopped moving forward when the pressure reached the set value 1. Then, the pressure was stabilized around -34 kPa during the pad stopped moving. The pressure began to decrease gradually again when the pad started moving backward, eventually the end-effector started moving forward when the pressure reached the set value 2. Even when the end-effector started moving, the pressure continued to decrease up to -41 kPa, because there was a delay before the end-effector started moving after detecting the set value 2, and the speed of the pad was slightly faster than that of the end-effector. When the pad and end-effector stopped moving, there was another change of pressure in the pad, because the pad was controlled by a relay circuit and a DC motor while the motion of the end-effector were caused by the manipulator. The moving speeds of the pad and the end-effector were 37.7 mm/s and 31.7 mm/s respectively. Therefore, the pad and the end-effector were unlikely to

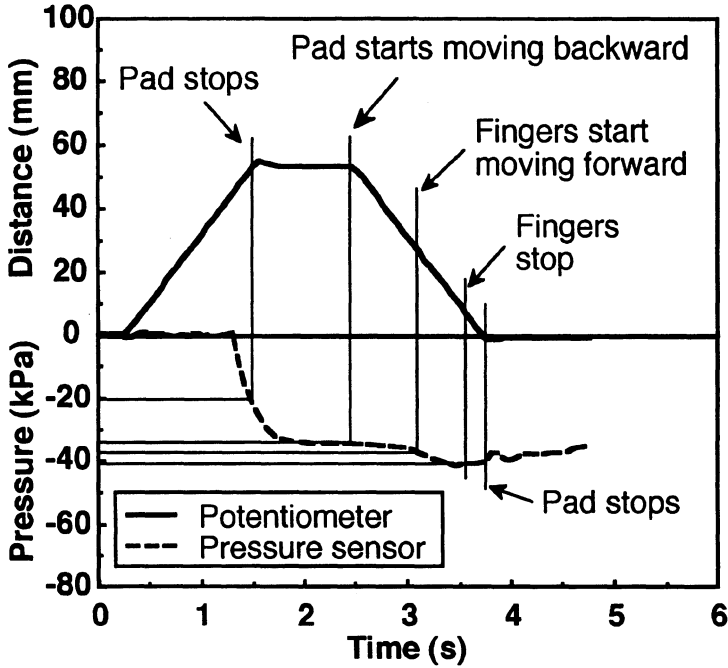


Figure 10. Relation between moving distance and pressure on fruit harvesting experiment.

stop exactly at the same time. After the pad starting moving backward, the pressure changed more slowly than that shown in Figure 9, because the fruit position moved toward the end-effector due to stem and peduncle movements when the pad pulled the fruit, however the fruit position was fixed in the case of Figure 9.

Twenty-three tomato fruits including immature, mature and over-mature fruits were used for the harvesting experiment. From the results of the experiment, the end-effector could harvest 91% of the fruits. It was observed that all the mature fruits were harvested successfully with no damage due to the added functions of these sensors. Fruits with a short peduncle were also harvested without separation from the pad. One immature fruit with rugged surface was not sucked by the pad because a clearance was left between the fruit and the pad. Some of the over-mature fruits were picked up not at peduncle joints but at calyxes, because their peduncle joints were hard to be bent. Even for some fruits which are partially hidden behind leaves, stems and other fruits, they could be successfully harvested, as long as they had exposed parts accessible by the pad, due to the extended moving distance of the pad. In this training method of tomato plant, it was difficult for the robot to harvest a fruit

which was completely hidden behind other fruits, leaves and stems, because they could be the obstacles for the approach of the end-effector. Therefore, it was considered that the robot could work more easily and the percentage of success of harvesting could be further improved if the training method was changed, for example the high density STTPS (Single Truss Tomato Production System) [13] in which only a single cluster is grown in a tomato plant and there are very few obstacles around each cluster.

The time required for harvesting one fruit was around 15 sec, from the initial posture of the manipulator to the release of a fruit. The time can be shorter by increasing the speeds of manipulator and DC motor on the end-effector.

5. Conclusion

In this research, the first prototype end-effectors with two plate fingers and the suction pad, and the improved second prototype were manufactured based on the physical properties of tomato plant and tested after they were attached to a manipulator. From the experiments described above, the followings were obtained.

1. The suction pad was an effective function to singulate a target fruit from the others in the same cluster.
2. The fruits with a short peduncle were harvested with no damage due to the functions of the second prototype end-effector to detect the pressure and the position of the suction pad.
3. The fruits which were partially hidden behind the obstacles were harvested by extending the stroke of the pad, though it was difficult for the robot to harvest the fruit which was completely hidden.
4. It was considered that the robot could work more easily and efficiently when the new training method was introduced to the tomato production system.

References

1. Fujiura, T., Ura, M., Kawamura, N. & Namikawa, K. (1990). Fruit Harvesting Robot for Orchard. *Journal of the Japanese Society of Agricultural Machinery* 52(1): 35–42.
2. Kondo, N., Monta, M., Shibano, Y., Mohri, K., Yamashita, J. & Fujiura, T. (1992). Agricultural Robots (2): Manipulators and Fruits Harvesting Hands. In *ASAE Paper*, No. 923518.
3. Ting, K. C., Giacomelli, G. A. & Shen, S. J. (1990). Robot Workcell for Transplanting of Seedlings (Part 1). *Trans. ASAE* 33(3): 1005–1010.
4. Ting, K. C., Giacomelli, G. A., Shen, S. J. & Kabala, W. P. (1990). Robot Workcell for Transplanting of Seedlings (Part 2). *Trans. ASAE* 33(3): 1013–1017.
5. Monta, M., Kondo, N., Shibano, Y., Mohri, K., Yamashita, J. & Fujiura, T. (1992). Agricultural Robots (3): Grape Berry Thinning Hand. In *ASAE Paper*, No. 923519.

6. Kawamura, N., Namikawa, K., Fujiura, T. & Ura, M. (1984). Study on Agricultural Robot (Part 1). *Journal of the Japanese Society of Agricultural Machinery* **46**(3): 353–358.
7. Okamoto, T., Shirai, Y., Fujiura, T. & Kondo, N. (1992). *Intelligent Robotics*. Jikkyo Shuppan, Japan: Tokyo.
8. Kondo, N., Monta, M., Shibano, Y. & Mohri, K. (1993). Basic Mechanism of Robot Adapted to Physical Properties of Tomato Plant. *Proceedings of International Conference for Agricultural Machinery and Process Engineering* **3**: 840–849.
9. Kondo, N., Monta, M., Fujiura, T., Shibano, Y. & Mohri, K. (1994). Control Method for 7 DOF Robot to Harvest Tomato. *Proceedings of the Asian Control Conference* **1**: 1–4.
10. Kondo, N., Monta, M., Shibano, Y. & Mohri, K. (1993). Two Finger Harvesting Hand with Absorptive Pad Based on Physical Properties of Tomato. *Environ. Control in Biol.* **31**(2): 87–92.
11. D'Esnon, A.G., Rabatel, G., Pellenc, R., Journeau, A. & Aldon, M. J. (1987). MAGALI – A Self Propelled Robot to Pick Apples. *ASAE Paper*, No. 871037.
12. Reed, J. N., He, W. & Tillett, D. (1995). Picking Mushrooms by Robot. *Proceedings of International Symposium on Automation and Robotics in Bioproduction and Processing* **1**: 27–34.
13. Ting, K. C., Giacomelli, G. A. & Fang, W. (1993). Decision Support System for Single Truss Tomato Production. *Proceedings XXV CIOSTA-CIGR V Congress*: 70–76.

A Computer Vision Method for Determining Length of Cheese Shreds

H. NI and S. GUNASEKARAN

*Biological Systems Engineering Department, University of Wisconsin, Madison, USA
(E-mail: guna@facstaff.wisc.edu)*

Abstract. A computer-vision based system was used to obtain images of shredded cheese. The images were processed by morphological transformation algorithms such as dilation and erosion to smooth the image edge contours. The smoothed image was skeletonized. Cheese shred lengths were determined from skeletonized images using syntactic methods. This method was successful in recognizing individual shreds even when shreds were touching or overlapping. Shred lengths calculated from the processed images compared very well with those measured manually.

Key words: image processing, thinning, skeletonizing, morphology, overlapping

Introduction

Most of the cheese used as a food ingredient is in one of the following machined forms: diced, shredded, grated or sliced. With numerous new cheese-containing food preparations introduced in the market place, use of machined cheeses will continue to grow. In 1993, shredded cheese was the fastest growing of all categories of the natural cheese market (about 10%), accounting for nearly one billion dollars in sales. Shredded natural cheeses are one of the primary outlets for the major varieties of cheese in all segments of food markets: food service, industrial and retail. It may also become an important outlet for many specialty varieties. This form of cheese was one of the few cheese categories which showed growth of sales in supermarkets over the past several years (Ruland 1993).

Much of the shredded cheese market is based on the integrity and uniformity of the shreds. In addition to being appealing to the eyes, shreds that retain their integrity melt uniformly and are easier to sprinkle on foods (Dubuy 1980). These attributes enhance sales and use of shredded cheese as a food ingredient. Machined cheeses make portion control and/or fill-weight control easy (Andres 1983). To take full advantage of the growing shredded cheese market, therefore, it is essential to offer cheese shreds that do not mat and/or fragment. However, cheese processors often find it difficult to maintain the

integrity of cheese shreds, especially when composition and manufacturing parameters vary widely. The trend to lower fat and nonfat cheeses has imposed additional, unresolved processing technologies in shredding cheese.

In order to select parameters that will assure good shredding characteristics, manufacturers routinely evaluate the quality of the shreds. However, the current evaluation method involves sieving a sample of shredded cheese to collect fragments that pass through a certain sieve size. This method, while focusing on fragmented small pieces, ignores evaluating the more important unbroken shreds. High quality shredded cheese has individual shreds of uniform size. Trying to evaluate individual shred characteristics manually is tedious and time-consuming. Therefore, cheese processors need a tool and a rapid method to evaluate individual shred size and shape characteristics with little human intervention. Apostolopoulos and Marshall (1994) recently began to exploit a fast, quantitative method to determine cheese shredability using computer image analysis techniques. However, their method was based on an unrealistic assumption that shreds did not touch or overlap.

The objectives of our work were to: 1) develop an image processing algorithm to analyze the image of cheese shreds without manually separating them and 2) quantify morphological features to characterize the lengths of individual cheese shreds.

Methods

Image acquisition

Shredded Mozzarella cheese samples were purchased from a local grocery store and used in the experiments. Three example cases were considered to develop the image processing algorithm: 1) individual shreds, 2) shreds touching each other, and 3) shreds overlapping each other. The shreds were arranged physically to represent these categories (Figure 1).

A visual scene of cheese shreds was acquired in computer through a CCD camera (Sanyo Model VDC 3874). High frequency noise in the image, apparently caused by lighting and imaging conditions, was removed by applying a digital low pass filter. By properly choosing the threshold value (by examining the image histogram), the image was binarized in which white pixels (with numerical value 1) indicated the objects (shreds) and dark pixels (with numerical value 0) indicated the background (Figure 2).

Image processing

An important approach in representing an object's structural shape is to reduce it to a graph. The process of thinning the binary image was selected to obtain

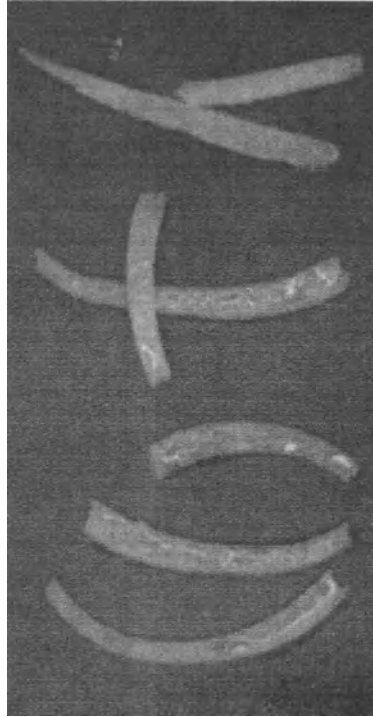


Figure 1. Image of cheese shreds.

its skeleton A skeleton is a line-thinned caricature of the binary image that summarizes the shape and conveys information about its size, orientation and connectivity (Gonzalez and Woods 1992). The following mathematical, morphological thinning operations, which guarantee the connectivity of skeleton, were used.

Dilation and erosion

Dilation and erosion are mathematical transformations that are dual to each other. By iteratively using dilation accompanied by erosion, we can smooth the contour, eliminate small holes and fill gaps in the binary image This dilation and erosion process is an important preprocessing step for obtaining a correct skeleton. This process changes the shred's contour the shred a little bit. However, considering that a shred's length is much larger than its width, preprocessing did not affect the shred skeletons significantly. The result of thinning the binary image in Figure 2 with dilation and erosion preprocessing



Figure 2. Binary image of cheese shreds.

is shown in Figure 3. The same thinning operation without preprocessing is shown in Figure 4. The circular loop in Figure 4 was caused by a noise component, the small hole in Figure 2. In Figure 3, the loop is not seen; it was filled by the preprocessing step.

Thinning

The process of thinning, sometimes called skeletonizing, can be implemented by a set of morphological operations. Thinning image A with a structural element B can be interpreted as sliding B over A. If a pixel of A has the same neighborhood value (in a 3×3 window) as that of B, then set that pixel of A to zero. The results of thinning the binary image of Figure 2 are shown (using binary image) in Figure 3 and (using numerical values) in Figures 5 and 6.

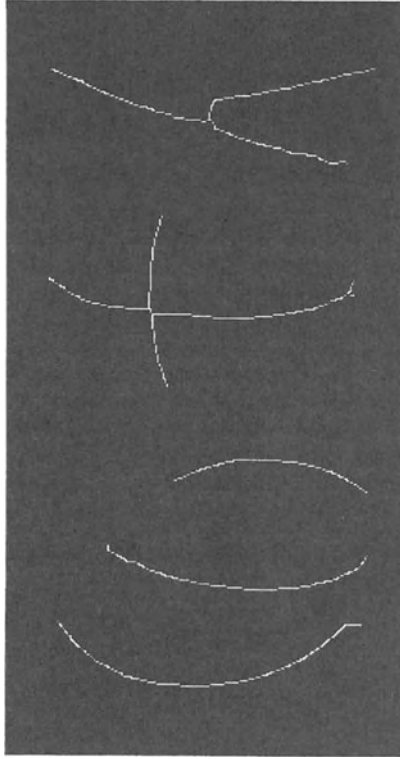


Figure 3. Thinned image of cheese shreds with preprocessing step.

Length measurement method using syntactic networks

One effective representation of objects in an image is modeling. Modeling can be interpreted as a syntactic pattern recognition approach. In syntactic graph, objects are transformed into primitives (represented by nodes) and their spatial relationships (represented by arcs). Primitives are usually chosen to be simple and to contain very little structural information. The approach used essentially consisted of two parts. The first part was to decompose a complex-shape and transform an object into a set of primitives (words) and their spatial relationships. The second part was to merge the primitives that satisfy certain pre-specified spatial constraints (grammatical rules). Our aim was to: 1) break the complex curves into a set of primitive segments, 2) calculate the length and orientation of each primitive segment and 3) merge the primitives together to obtain the length of individual shreds. The process of forming a syntactic graph is described as follows.

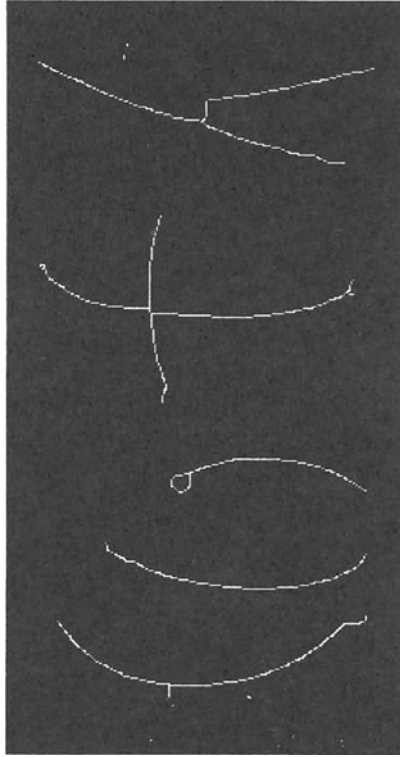


Figure 4. Thinned image of cheese shreds without preprocessing step.

Tracing the primitive segment

After thinning, there are only three kinds of pixels in the image. They are:

- **terminal point** – the pixel at the beginning or at the end of an arc;
- **hinge point** – the pixel where the arc changes its orientation or where several arcs join together;
- **transition point** – the pixel that is neither a terminal point nor a hinge point.

In order to detect terminal or hinge points in an image, a 3×3 summation filter was used over all image pixels. A summation filter can be interpreted as counting the number of neighbor pixels with a value of 1 in the 3×3 window. The summation filter returns a value of 2 for terminal points and a value greater than 3 for hinge points. An example of pixel counting for images in Figure 6 is shown in Figure 7. Pixels with value 2 are terminal points; those with value 4 are hinge points; and those with value 3 are transition points. To obtain a

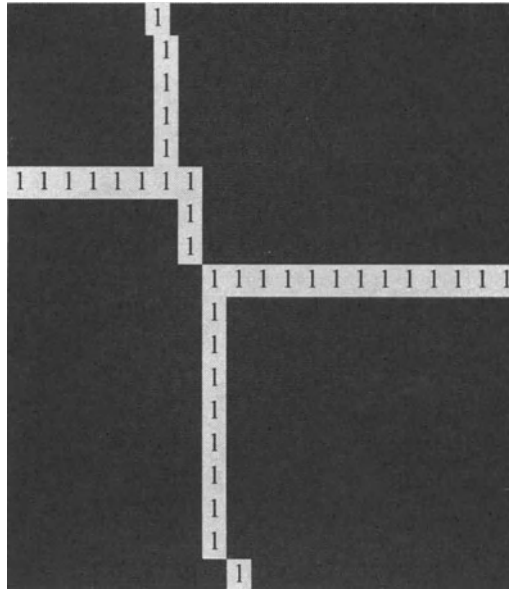


Figure 5. Result of thinning a binary image of two overlapping shreds (1's represent shreds).

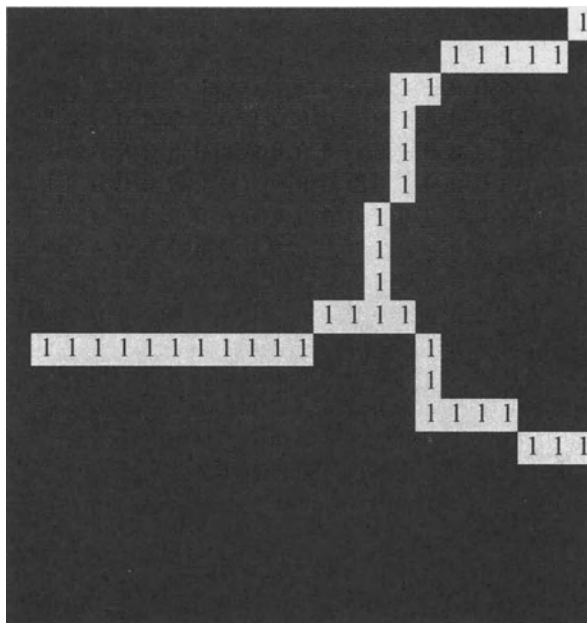


Figure 6. Result of thinning a binary image of two touching shreds (1's represent shreds).

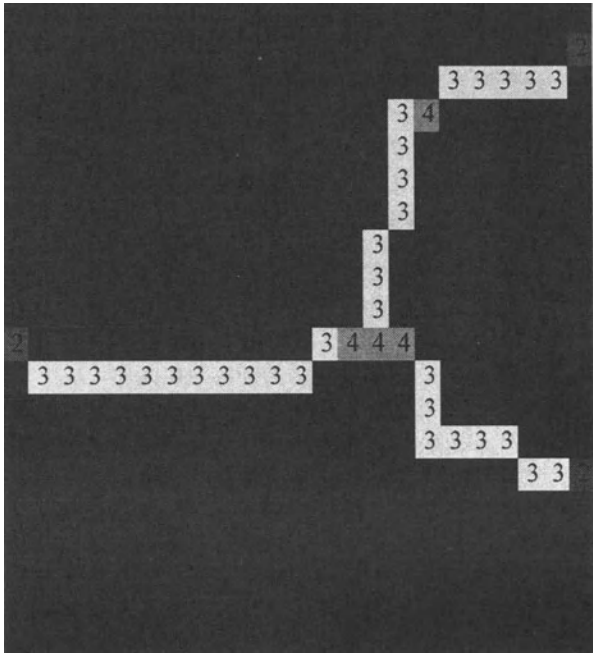


Figure 7. Identifying terminal, hinge, and transition points.

Table 1. Primitive segments of Figure 7

Primitive segment	Length (pixels)	Starting point (column, row)	Ending point (column, row)
1	14	(0, 9)	(13, 9)
2	10	(22, 5)	(15, 9)
3	7	(22, 18)	(16, 16)
4	8	(15, 16)	(14, 9)

primitive segment, we started from a terminal point and traced the curve until meeting another terminal point or hinge point. The pixel's terminal or hinge points formed a primitive segment. For each primitive segment, information about the location, length and orientation was recorded. To avoid repeatedly tracing the same curve, a pixel was set to 0 after it had been visited. A result of forming the primitive segments of Figure 7 is shown in Table 1.

Table 2. Results of combining the primitive segments in Table 1

Primitive segment	Length (pixels)	Starting point (column, row)	Starting point (column, row)
1	24	(0, 9)	(22, 5)
2	0	(0, 0)	(0, 0)
3	15	(22, 18)	(14, 9)
4	0	(0, 0)	(0, 0)

Combining the primitive segments

All the primitive segments (words) obtained (shown in Table 1) were merged according to the following grammatical rules.

- If two primitive segments meet at a point and their orientations are similar, then merge two segments and update the starting and ending points. Next, delete the old primitive segments. In determining the similarity of orientations, the closeness of the clockwise angle made by the primitive segments with the horizontal was used as the criterion. Theoretically, this method may lead to a false combination of two unrelated primitives. However, we feel that such occurrences are uncommon and can be accounted for by taking larger sample sizes.
- If two primitive segments meet at a point and only two segments appear at that point, then merge these two primitive segments and update the starting and ending points. Next, delete the old primitive segments.

We iteratively merged those segments that satisfied the grammatical rules until there were no further primitive segments to be merged. Results of combining the primitive segments in Table 1 are shown in Table 2.

Results and Discussion

The algorithm was tested by processing two test images formed by manually placing shreds to represent all three cases considered in developing the algorithm (i.e., single, touching and overlapping shreds). Test image 1 had 4 shreds; and test image 2 had three shreds. The thinned version of the test images are shown in Figures 8 and 9. The shred lengths were computed both by processing them using the algorithm developed and by manually measuring with a micrometer. When necessary, curved shreds were broken into two or three linear segments to facilitate manual measurements. The calculated and measured shred lengths are presented in Tables 3 and 4.

The error in shred length measurement was as low as 0.2% and not more than 10%, even in the worst cases. This is evidence that the algorithm was able

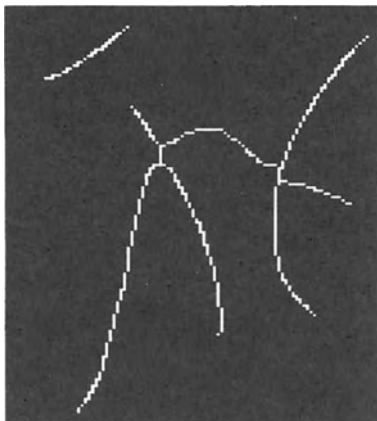


Figure 8. Test image 1.

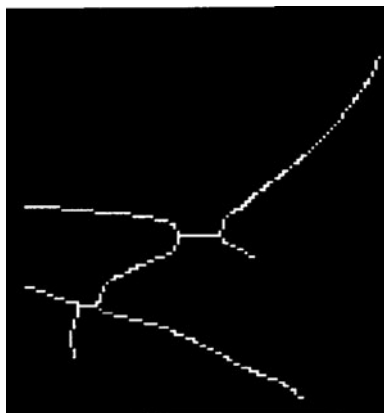


Figure 9. Test image 2.

to recognize individual shreds fairly well, even when they were touching and overlapping. Thus, it offers a practical means for cheese shred manufacturers to routinely evaluate the integrity of their product so as to improve the overall quality.

Conclusions

An image processing algorithm was developed to recognize individual shreds and automatically measure the shred length. Three cases of shred layout were

Table 3. Calculated and measured lengths of shreds in test image 1 (Figure 8)

Shred number	Calculated length		Measured length (mm)	Error (%)
	Pixels	mm		
1	148	64.1	65.0	1.4
2	77	33.3	32.0	4.1
3	102	44.1	44.0	0.2
4	34	14.8	16.0	7.5

Table 4. Calculated and measured lengths of shreds in test image 2 (Figure 9)

Shred number	Calculated length		Measured length (mm)	Error (%)
	Pixels	mm		
1	121	52.4	54.0	3.0
2	64	27.7	31.0	7.4
3	82	35.4	39.0	9.2

tested: 1) single shreds, 2) two shreds touching each other and 3) two shreds overlapping each other. In all cases, the algorithm performed very well. The error in shred length measurement was as low as 0.2% (and not more than 10%).

References

- Andres, C. (1983). Natural Cheese in New Form Provides Improved Aesthetic and Functional Qualities. *Food Processing* 44(12): 64–66.
- Apostopoulos, C. & Marshall, R. J. (1994). A Quantitative Method for Determination of Shred Quality of Cheese. *J. of Food Quality* 17: 115–128.
- Dubuy, M. (1980). The French Art of Shredding Cheese. *Food Proc. Industry* 49: 52–53.
- Darwish, A. M. & Jain, A. K. (1988). A Rule Based Approach for Visual Pattern Inspection. *IEEE Trans. on PAMI* 10(1).
- Gonzalez, R. C. & Woods, R. E. (1992). *Digital Image Processing*. Addison-Wesley Publishing Corp.
- Harlick, R. M., Sternberg, S. R. & Zhang, X. (1987). Image Analysis Using Mathematical Morphology. *IEEE Trans. on PAMI* 9(4).
- Jang, B. K. (1990). *Shape Analysis Using Mathematical Morphology*. Ph.D Thesis, University of Wisconsin-Madison, Madison, WI.
- Lam, L. & Lee, S.-W. (1992). Thinning Methodologies – a Comprehensive Survey. *IEEE Trans. on PAMI* 40(9).
- McDonald, T. & Chen, Y. R. (1990). Application of Morphological Image Processing in Agriculture. *Trans. of the ASAE* 33(7).
- Maragos, P. & Schafer, R. W. (1986). Morphological Sskelton Representation and Coding of Binary Image. *IEEE Trans. on ASSP* 34(5).
- Ruland, S. (1993). Rough Going. *Dairy Field* 176(8): 53–56.
- Sinha, R. M. K. (1987). A Width Independent Algorithm for Character Skeleton Estimation. *Computer Vision, Graphics and Image Processing* 40: 388–397.

Automated Modelling of Physiological Processes During Postharvest Distribution of Agricultural Products

M. SLOOF

Artificial Intelligence Group, Vrije Universiteit, Amsterdam and Agrotechnological Research Institute (ATO-DLO), Wageningen, The Netherlands (E-mail: M.Sloof@everest.nl)

Abstract. In this paper, we present an approach to automated modelling of physiological processes occurring during postharvest distribution of agricultural products. The approach involves reasoning about the reuse of both qualitative and mathematical models for physiological processes, and constructs quantitative simulation models for the postharvest behaviour of agricultural products. The qualitative models are used to bridge the gap between the modeller's knowledge about the physiological phenomenon and the mathematical models. The qualitative models are represented by *knowledge graphs*, that are conceptual graphs containing only causal relations, aggregation relations, and generalisation relations between domain quantities. The relationships between the mathematical models and the qualitative models are explicitly represented in *application frames*.

The automated modelling task consists of two subtasks. In the first subtask, *Qualitative Process Analysis*, a process structure graph is constructed using the qualitative models as building blocks. The process structure graph is a qualitative description of the phenomenon under study, that contains the processes that are responsible for the behaviour of the phenomenon. The process structure graph serves as a focus for the second subtask, *Simulation Model Construction*. This subtask uses a library of mathematical models to compose a quantitative simulation model that corresponds to the process structure graph constructed in the first subtask. The approach is illustrated with the construction of a model for the occurrence of chilling injury in bell peppers.

Key words: automated modelling, postharvest physiology, simulation models

1. Introduction

During postharvest storage and distribution of agricultural products loss of quality may occur. Therefore, activities such as cooling and packing, are performed to minimise this quality loss. To predict the effects of a distribution chain on the quality of a product, quantitative simulation models, called quality change models, are used.

We previously presented a conceptual model in which quality change of agricultural products depends on three factors: on the product itself, on the needs and goals of the user, and on the market situation at the moment of purchasing the product [1]. The observed product behaviour is the result of interactions between physiological, chemical and physical processes occur-

ring in the product. For these processes, separate mathematical models can be developed. These individual models can then be reused to compose quality change models for different products and for different distribution chains.

Before such simulation models can be constructed, it has to be determined which aspects of the postharvest behaviour have to be incorporated in the simulation model. This analysis phase is often left to the modeller. Many modelling environments only present a library of submodels to the user, who must then select the appropriate submodels. These modelling environments do not aid the modeller in analysing the phenomenon at hand.

We have developed a knowledge-based system to support a modeller in reusing a library of mathematical models for physiological processes to construct quantitative simulation models for the postharvest behaviour of agricultural products. This system is called DESIMAL, which is an acronym for “DEsign and Specification of Interacting MAThematicaL models” [2, 3].

This article presents the approach to automated modelling as applied in the DESIMAL system. The organisation of this article is as follows. In Section 2 we review some approaches to modelling support and to automated model construction. In Section 3 we introduce the phenomenon that is used as a running example to illustrate the DESIMAL approach: the occurrence of chilling injury in bell peppers. In Section 4 we describe the representation of the knowledge in the library used in our DESIMAL method. This library contains both qualitative and mathematical knowledge about physiological processes, and contains explicit applicability knowledge to connect these two description levels. These models are the building blocks for the two modelling tasks used in DESIMAL. These modelling tasks are described in Section 5.

2. Related Research

2.1. *Modelling support*

Building a simulation model for a physiological phenomenon under study requires that the modeller translates his knowledge about the physiological behaviour of the product into a set of mathematical equations. For different application domains, modelling support methods have been described in which such a large semantic gap between the expert’s knowledge about a problem and the library of building blocks is bridged by introducing one or more additional description levels. Below, we describe modelling support methods for three different application domains:

- The Knowledge-Based Model Construction method of Murray and Sheppard [4] supports a modeller in the construction of discrete event simulation models for queuing systems. First, a dialogue is held with the

modeller to acquire a specification of the model to be constructed. In this dialogue knowledge of the domain and general simulation modeling knowledge is used. Second, this model specification, general simulation modelling knowledge, and knowledge of the target language are used to construct a discrete event simulation model in the target language.

- The *evolutionary modelling* method of Top [5] emphasizes that modelling is a “process of making incremental and systematic assumptions”. In this method four description levels are identified in models for physical systems. First, the modeller assumes a decomposition into a set of *functional components*. Second, assumptions are made about the *physical processes* occurring in the functional components. Third, *mathematical equations* are assumed for each process leading to a mathematical model for the complete system. Fourth, values for the model parameters are assumed, which is the *model data* level. Each of these four description levels describes a different aspect of the system under study, that is not captured in the other levels. Hence, a complete model for that system contains descriptions for all four levels.
- Mili [6] describes a framework for program library documentation that provides three documentation templates: (1) for the problem to be solved, (2) for the programs in the library, and (3) for the applicability of a program to solve a particular problem. The third documentation template serves as an intermediate description level between the problem to be solved and the available programs. It describes when a program is suitable for a problem, and how the program must be instantiated for that problem. It also describes when and why a program is not suitable to a problem and then points to other programs in the library that may be suitable. The framework is applied to document financial and mass spectrometry libraries.

These methods illustrate different approaches to modelling support: Murray and Sheppard use a dialogue to construct a specification of the simulation model to be constructed. The information gathered in this dialogue is, however, not stored in a library. Top introduces two intermediate description levels between the expert’s knowledge about a physical system and the mathematical model for that system. The library contains separate building blocks for each description level. Each building block is related to other building blocks in the adjacent description levels [7]. The focus of Mili’s research is “mainly on helping users understand the usefulness and the limitations of the programs used rather than on helping the users to select the correct program.” This is manifested by the application templates in the library that explicitly specify a mapping between the programs in the library and the problems to be solved.

The additional description levels introduced in these approaches serve to make explicit the assumptions and decisions that a modeller makes in developing a model. Explicit representation of modelling assumptions and reasoning about these assumptions is the key to automated model construction.

2.2. Automated model construction

In the field of qualitative reasoning [8, 9], several approaches to automated model construction have been presented. The automated modelling approaches construct so-called scenario models for answering queries about a given system. The *query* usually consists of three parts: (1) a set of quantities of interest, that have to be explained by the scenario model, (2) a structural description of the system under study, consisting of the components and their connections, and (3) a specification of the initial state of the system.

From this query a *scenario model* is constructed using a library of model fragments. The scenario model must be the simplest model that is sufficient to explain or predict the behaviour of the quantities of interest specified in the query. Several approaches have been described:

- The *compositional modelling* method of Falkenhainer and Forbus [10, 11] provides basic concepts to automated construction of scenario models for the analysis of a system's short-term behaviour using a library of model fragments.
- Iwasaki and Levy [12] describe an approach to automated construction of models for *simulation*. Given a query with the above structure, the constructed scenario model must explain how the quantities of interest change over time. As the simulation may go through any state satisfying the initial state specified in the query, the scenario model must contain all model fragments that can be active in any of each of these states, whether they are actually reached or not. A model fragment is active in a system state, if the input conditions and the operating conditions of the model fragment are satisfied in that state. The active model fragments form a simulation model that determines the next state of the system. This method differs from the method of Falkenhainer and Forbus, in that the scenario models contain all submodels that are possibly reachable from the initial state. Part of the model selection task is performed during the simulation experiments to derive the applicable submodels at each state of the system.
- Nayak [13] describes a method for automated model construction that aims at explaining a user specified causal relation in a system under study. The causal relation specifies the behaviour that the user is interested in, for example the heat produced by a current through an electrical wire.

- Rickel's TRIPEL system [14, 15] constructs qualitative models in the domain of plant physiology. This system uses a large botany knowledge base, containing causal relations between plant properties to specify physiological processes, and encapsulation relations on plant properties and on causal relations to express differences in levels of detail. Because many of the relationships in the botany knowledge base as yet cannot be modelled quantitatively, the level of detail reached in the qualitative models produced by Rickel's system cannot yet be reached in quantitative models.

In the rest of this section the main characteristics of these methods are described. We describe the contents and organisation of the model fragments, and we describe how the different methods meet the requirements of sufficiency and simplicity of the constructed scenario models. More extensive reviews and discussions of approaches to automated model construction can be found in [16, 17].

2.2.1. *Contents and organisation of model fragments*

Except for Rickel's approach, that uses a large knowledge base of causal relations between plant properties, the automated modelling approaches use a library of *model fragments*. Generally, a model fragment is a partial specification of the behaviour of an aspect of a system's behaviour, so that for a complete description of that aspect several model fragments are needed.

The specification of a model fragment contains both the formulation of the model fragment and the assumptions underlying the formulation. Falkenhainer and Forbus define a model fragment as consisting of four parts:

- *Participants* are the entities in the domain to which the model fragment applies. The participants are subject to conditions that define the *structural configuration* of the participants. The model fragment is only applicable to entities that satisfy these conditions.
- *Operating conditions* are restrictions on the values of the variables in the formulation of a model fragment. The operating conditions are used during the simulation to determine whether the model fragment can be activated, and to determine whether the values of the variables are valid.
- *Underlying assumptions* specify for which queries a model fragment may be relevant, and specify decisions about the formulation of the model fragment, that were made during modelling. The underlying assumptions correspond to the *model selection heuristics* in [18]. The latter term clearly indicates that these assumptions are used when selecting an appropriate model fragment for a component or phenomenon when more than one model fragment applies.

```

ModelFragment ContainedLiquid( cl )
  Participants
    can Conditions Fluid-container( can )
    cl Conditions Contained-liquid( cl ) And
      Container-of( cl, can ) And
      Substance-of( cl, sub )
  Operating-Conditions
  Assumptions
    Consider( Contained-Fluids( can ) )
  Behaviour
    level(cl) = ( 4 * mass(cl)) / ( density(sub) * PI * square(diameter(can)))
    pressure(bottom(can)) = level(cl) * density(sub) * G

```

Figure 1. A model fragment defining the level and the pressure of a liquid in a container. Adapted from Figure 2 in [11].

- *Behaviour relations* are the qualitative or quantitative relations imposed by the model fragment between physical quantities of the participants, and can thus be seen as the formulation of the model fragment.

Figure 1 displays a model fragment that defines some properties of a liquid *cl* contained in a can. The conditions in the Participants section specify the structural configuration for which the behavioural relations are valid. The Consider statement specifies that the model fragment is relevant for queries about cans containing a fluid.

Model fragments that describe one aspect with differing underlying assumptions represent different modelling decisions on that aspect, and thus cannot be used at the same time to describe the aspect. Such alternative model fragments (and the corresponding underlying assumptions) are grouped into *assumption classes*. In the construction of a scenario model, for each relevant aspect of the system one model fragment must be selected (hence a modelling decision must be made) from the assumption class of that aspect. For example, a scenario model for a battery may take into account two aspects: (1) the voltage of the battery, and (2) the charge-level of the battery¹. Each aspect has a separate assumption class containing the alternative model fragments for that aspect. The alternatives for the voltage are a constant voltage, or a charge-sensitive voltage. The alternative model fragments for the charge level describe a constant charge-level, a normal accumulation recharge, or an accumulation with aging. Hence, a complete model for the battery needs one model fragment from the voltage assumption class and one model fragment from the charge level assumption class.

2.2.2. *Sufficiency and simplicity of the scenario models*

The automated modelling approaches impose requirements of sufficiency and simplicity on the constructed models.

The requirement of *sufficiency* or *adequacy* means that the scenario model only includes those aspects of the system under study that are needed to analyse the situation specified in the query with an appropriate level of detail. The consequence of the sufficiency requirement is that as soon as the query changes, a new scenario model must be composed. The approaches differ on the implementation of the sufficiency requirement:

- Falkenhainer and Forbus use a strict structural part-of hierarchy of all objects in the domain to determine the system boundary. Starting from the required scenario elements, the part-of hierarchy is traversed upwards, until the smallest system is found that comprises all required scenario elements. The objects in the domain outside this smallest system need not be considered.
- Iwasaki and Levy use knowledge about the relevancy of the model fragments to determine which model fragments are required in the scenario models. A model fragment is relevant if, in some state of the system, any of the quantities mentioned in the input or in the operating conditions of the model fragment has a causal effect on any of the quantities mentioned in the query. Hence, the scenario model contains all model fragments that may be needed to describe a piece of behaviour in some state of the system that is reachable from the initial state in the query.
- Nayak uses a function-based approach. The scenario model must explain the function of the device that is of interest. This function is specified in the query as an input/output relation, called an *expected behaviour*. Only the model fragments on this causal path are relevant. For example, a wire can be modelled as an ideal conductor, as having a certain elasticity, or as a thermal-resistor giving off heat. Depending on the expected behaviour, the appropriate viewpoint is selected.
- Rickel uses knowledge about the time scales at which changes in quantities occur. Only those influences are considered to be relevant that have a time scale equal or smaller than the time scale of interest. Thus, if the quantities in the query change at a time scale of minutes, then the changes that take hours or days are considered to be negligible.

The requirement of *simplicity* ensures that the models describe the behaviour at the lowest level of detail that is needed to answer the query. The model fragments are ordered by a *simpler-than* relation based on the underlying assumptions of the model fragment. For example, a model for a chemical reaction assuming first-order kinetics is simpler than a model that assumes Michaelis-Menten kinetics. Usually, the simplest model fragment is chosen,

until it proves unsuitable to describe the behaviour. Nayak first selects the most complex model fragments, and then applies simplifications to find the simplest sufficient scenario model. Whereas Falkenhainer and Forbus order the model fragments in an assumption class increasing complexity, and thus use an implicit measure of simplicity, Iwasaki and Levy base the measure of simplicity on the relevance of quantities, which is explicitly represented as *relevance claims*. A model fragment that assumes fewer quantities to be relevant is simpler. This *simpler-than* relation corresponds to the relations between models in the Graphs of Models approach of Addanki [19].

3. Example: Chilling Injury

In the rest of this article, we present our DESIMAL approach to automated modelling of physiological processes occurring during postharvest distribution of agricultural products. As a running example we use the complex phenomenon of chilling injury.

Chilling injury is a general term for visible forms of damage that may occur when products are stored at too low temperatures. The injury normally appears after the chilling period, when the product may already be stored under optimal conditions. This delayed appearance makes chilling injury difficult to model. By decomposing the phenomenon of chilling injury into generic processes and behaviour patterns, Tijskens et al. have developed a simulation model to predict the occurrence of chilling injury in cucumber fruits and bell peppers [20]. The decomposition was based on the following assumptions:

- Chilling injury is known to be the visible effect of free radicals that emerge from cells with damaged cell membranes. This complex process was assumed to be autocatalytic with respect to the free radicals and was modelled as such. Two quantities are affected that are of interest in the model: the amount of chilling injury, and the concentration of free radicals.
- Under normal conditions chilling injury does not occur, hence, the free radicals must be removed or inactivated in some way. This radical scavenging process was assumed to be an enzymatic process. This process is influenced by the concentration of free radicals and by the enzyme concentration. As the enzyme concentration in an enzymatic process remains constant, the radical scavenging process only affects the concentration of free radicals.
- The enzyme in the radical scavenging process was assumed to denature at low temperatures, to account for the fact that chilling injury only occurs after a period of too low temperatures. As the denaturation is irreversible,

only the effect on the enzyme concentration is relevant, and the changes in the concentration of the inactivated enzyme can be ignored.

Hence, by making the above assumptions and by using generic processes and patterns of behaviour, it proved possible to develop a quantitative simulation model for the complex phenomenon of chilling injury. This model also correctly explained chilling injury phenomena that were not accounted for in the development of the simulation model, which proves the validity of the approach.

4. Representation of the Domain Knowledge

The DESIMAL method constructs quantitative models for simulation of the changes over time of postharvest physiological processes occurring in agricultural products. These quantitative models are composed from a library of mathematical models.

The model fragment specification used in the automated modelling approaches described in Section 2.2.1, contains two types of knowledge. The participants and the behaviour relations specify knowledge about the model formulation. The underlying assumptions and the structural configuration of the participants specify meta-level knowledge about the relevancy of the model fragments to questions of interest.

Our approach is to separate these types of knowledge into three separate knowledge levels, that together form the library used in the modelling tasks of the DESIMAL method:

- A *qualitative knowledge* level, consisting of qualitative models for physiological processes and for decompositions of aggregate quantities into subquantities. The contents and representation of the qualitative knowledge are presented in Section 4.1.
- A *mathematical knowledge* level, containing the mathematical models that are the building blocks for the quantitative simulation models. The specification of the mathematical models is discussed in Section 4.2.
- An *application knowledge* level between the qualitative and mathematical knowledge levels, that for each mathematical model specifies which qualitative models are described by the mathematical model. The contents and the representation of the applicability knowledge is discussed in Section 4.3.

4.1. Qualitative knowledge

To represent the qualitative knowledge in the DESIMAL library, we propose a representation formalism, that is based on conceptual graphs [21]. The qual-

itative models in the DESIMAL library represent how complex phenomena of quality change in agricultural products can be described as interactions between a number of predefined generic physiological processes. Qualitative models represent the influences imposed by the processes, but hide the quantitative details of these influences that are not needed to find an adequate decomposition of the phenomenon under study.

As a basis for the representation of these processes, *knowledge graphs* [22, 23] are used. A knowledge graph is a conceptual graph with a limited relation set. The doctrine of knowledge graphs is that one should limit the number of different relation types. This is in contrast to conceptual graphs or logical representations where the set of relation types is unlimited so that one can introduce new relation types whenever needed [21]. Each knowledge graph represents the knowledge about a subject in the application domain. The nodes in a knowledge graph are concepts that are relevant for the subject. The arcs between these nodes represent cause-effect, part-of, and kind-of relationships between domain concepts. The arcs in a knowledge graph are labeled with abbreviations of the relation types: causal relations are labeled with CAU, part-of relations are labeled with PAR, and kind-of relations are labeled with AKO. More complex conceptual structures can be formed by grouping concepts and relations, and assigning a name to it. Such a subgraph is called a *frame*. To explicitly specify that a concept belongs to a frame, FPAR relations are used. In the original knowledge graph formalism of [22, 23], frames are in turn concepts in the knowledge graph, and can be related to other concepts in the knowledge graph.

In DESIMAL, knowledge graphs are used to represent the postharvest physiological behaviour underlying the quality change of agricultural products. Each knowledge graph is a qualitative model for one physiological phenomenon in an agricultural product. The knowledge graphs together form a qualitative theory about postharvest behaviour of agricultural products.

The concepts in these knowledge graphs are the product quantities and external factors. Two types of frames are used: *process frames* to represent physiological processes and *decomposition frames* to specify decompositions of quantities. Below, these frames are elaborated. An example of a knowledge graph is presented in Section 4.1.4.

4.1.1. *Process frames represent physiological behaviour*

A *process frame* represents the behaviour of a physiological process at *one* level of detail. The arcs between the quantities in a process frame represent two types of causal relations:

- dcau(source, target)

The DCAU relation represents a differential causal relation, which means

that the value of the **SOURCE** quantity determines the rate of change (first derivative) of the **TARGET** quantity. DCAU relations correspond with influences in Qualitative Process Theory (QPT) [24]. The differential causal relations represent a differential equation for the target quantity.

- **fcou(source, target)**

The FCAU relation represents a functional causal relation, which means that the value of the **SOURCE** quantity determines the value of the **TARGET** quantity. FCAU relations correspond with qualitative proportionalities in QPT. The functional causal relations represent an algebraic equation for the target quantity.

4.1.2 *Decomposition frames relate different levels of detail*

The second type of frames, *decomposition frames*, relate different levels of detail. Each decomposition frame represents the decomposition of a quantity into its subquantities at a more detailed description level, and contains only PAR relations:

- **par(detailed, aggregate)**

The PAR relation represents that the **detailed** quantity is a part of the **aggregate** quantity. A PAR relation defines a connection between the different levels of detail on which the behaviour of the **aggregate** quantity can be described.

Causal relations are not allowed in a decomposition frame, because these frames do not represent postharvest physiological behaviour.

A decomposition frame represents that the aggregate quantity can be modelled either as a black box quantity disregarding the detailed subquantities, or as the composite of all the subquantities specified in the decomposition frame. Hence, decomposition frames provide the means to select the level of detail that is appropriate for describing the phenomenon under study.

Only decompositions of quantities can be represented. A decomposition of a complex process into subprocesses is modelled by introducing intermediate quantities that are influenced by the subprocesses. The quantity that is affected by the complex process is then a composition of the intermediate quantities.

4.1.3 *Frames are connected by the quantities*

A decomposition frame relates quantities at different levels of detail, but does not specify the behaviour of the quantities. Similarly, a process frame specifies a physiological process, but does not specify interactions between processes. To connect the frames in a knowledge graph, two types of relations are used at the level of the quantities in the frames:

- **equ(quantity1, quantity2)**

The EQU relation specifies that quantity1 in one frame is exactly the same as quantity2 in another frame. The EQU relations are used to connect

separate frames that together describe one complex physiological process. These separate frames then contain quantities, possibly with differing names, that represent the same entity in the physiological process. The EQU relation explicitly represents this relationship.

– ako(specific, generic)

The AKO relation represents that the specific quantity is (modelled as) a specialisation of – or a kind of – the generic quantity. An AKO relation can represent *taxonomic* knowledge valid in the whole domain, e.g. the quantity RadicalScavengingEnzyme is a kind of ActiveEnzyme. An AKO relation may also represent a *modelling assumption* for a specific quantity, e.g. the relation ako(Radicals,AutoCatalyst) in Figure 2 represents that the release of free radicals from the cell membrane is modelled as an autocatalytic process.

4.1.4. Example

Figure 2 displays the knowledge graph frames for the chilling injury phenomenon. The process frame ChillingProcess in this figure only contains the product quantities that are involved in the phenomenon. The relations between these quantities are specified in the frames these product quantities are AKO-related to.

The phenomenon of chilling injury is a complex interaction of four generic processes. Three of these processes involve the Radicals, as is represented by the AKO relations to generic process frames.

- The quantity Radicals is a specialisation of the generic quantity AutoCatalyst in the process frame AutoCatalysis to represent that the increase of the free radicals is modelled as an autocatalytic process.
- The quantities Radicals and ChillingInjury are connected to the quantities ConsumedReactant and ProducedReactant in the process frame ChemicalReaction to represent that the visible effect of free radicals on cell membranes (i.e. chilling injury) is modelled as a chemical reaction that consumes Radicals to produce ChillingInjury.
- The quantities RadicalScavengingEnzyme and Radicals are connected to quantities in the process frame EnzymaticReaction to represent that the radicals may be consumed (scavenged) in an enzymatic process. The RadicalScavengingEnzyme has an AKO relation to the fourth process in the chilling injury phenomenon, IrreversibleEnzymeDenaturation, to represent that the enzyme may denature irreversibly to an inactive form.

Figure 2 furthermore contains the elaboration of the process frame ChemicalReaction. This process frame has EQU relations to the separate process frames describing the degradation of the ConsumedReactant and

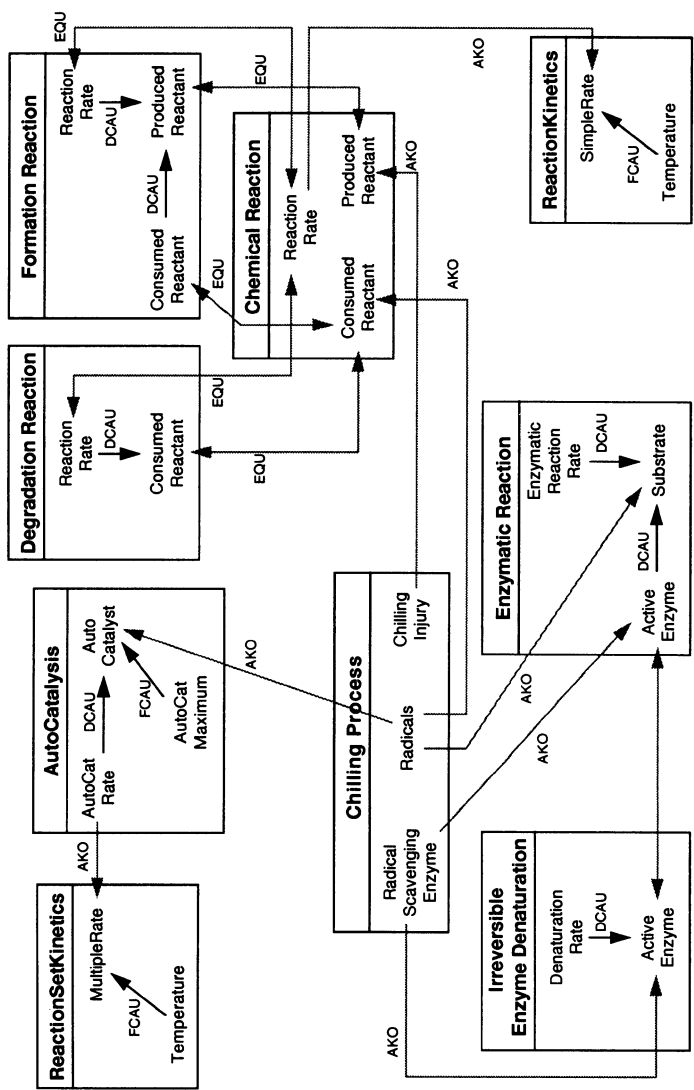


Figure 2. Knowledge graph showing how the occurrence of chilling injury can be modelled with generic processes.

the formation of the ProducedReactant. The EQU relations represent that the quantities in the three frames are the same, and thus, that the two reactions have a common ReactionRate. Separating the degradation and formation reactions enables the modeller to include only the effect on one of these quantities. More complex chemical reactions can be modelled by using decomposition frames for specialisations of ConsumedReactant and ProducedReactant.

4.2. *Mathematical models*

The second level in the DESIMAL library contains the mathematical models. The specification of a mathematical model consists of the formulation, the operating conditions, and the variables that appear in the formulation and in the operating conditions. Figure 3 displays the specifications of mathematical models for an enzymatic reaction with Michaelis-Menten kinetics.

- **formulation**

The mathematical equations in the formulation of a mathematical model must be specified in the order of computation. Each equation must be an assignment of the value of the right hand side expression of the equation to an internal or output variable of the submodel. An expression `integrate(diffeq)` may be used in an equation to specify that the value of the variable is the result of an integration of the differential equation `diffeq` over some small time step. The time step is determined by the numerical integration routine applied in the executable simulation model, and is not part of the specification of the mathematical model. The `integrate` expression in the second equation in the formulation of linear-MM in Figure 3 specifies the differential equation $dS/dt = -V_{max}$.

- **operating conditions**

The operating conditions are specified as an inequality between the variables of the mathematical model. The operating conditions must be satisfied before the model can be applied. As the values of the variables used in the operating conditions must be known to decide on the activation or deactivation of the model, the variables used in the operating conditions must also be specified in the `variables` section, as inputs of the mathematical model.

- **variables**

This section specifies the variables that appear in the formulation and in the operating conditions of the mathematical model. The variables are time series of simulated values, so that the construct `var(t)` can be used to refer to the value of the variable `var` at time `t`. This is, however, not encouraged, because the numerical integration routine that is applied in the executable simulation model may use a variable step size, and therefore may not calculate a value for the variable at the specified time

point. However, this construct can safely be used to refer to initial values of variables (at time $t = 0$), as is done in the formulation of linear-MM for the variables P and S.

In many model libraries, the models are formulated in equation form, allowing one model formulation in the model library for different causal orderings of the variables. The mathematical models for postharvest physiological behaviour are in general applied with one fixed causal ordering of the variables, so that an assignment form suffices. This does, however, not hold for models describing physical phenomena such as diffusive and osmotic transport, which also are important transport processes in agricultural products. For these processes the library must contain several mathematical models, one for each causal ordering.

4.2.1. *Example: models for enzymatic reactions*

Figure 3 displays the specifications of the mathematical models for an enzymatic reaction with Michaelis-Menten kinetics. The calculation of the variables Vmax and Km is separated from the models for the changes in the Substrate and the Product of the enzymatic reaction. This enables the modeller to use Vmax and Km as inputs of the simulation model, or to calculate the values of these variables within the simulation model.

4.3. *Applicability knowledge*

So far, two representation formalisms have been described for the postharvest physiological behaviour of agricultural products: a qualitative representation in terms of processes and decompositions of quantities (Section 4.1), and a quantitative representation in terms of mathematical equations (Section 4.2).

To connect these two representation formalisms, the DESIMAL library has an intermediate level consisting of *application frames*. An application frame relates a mathematical model to one or more qualitative knowledge graph frames. The application frames provide additional knowledge needed for automated construction of simulation models that is not present in the specifications of the mathematical models.

An application frame for a mathematical model specifies (1) a *process structure* consisting of the qualitative knowledge graph frames for which the mathematical model is applicable, (2) an *interpretation* that maps the variables of the mathematical model onto the quantities in the process structure, and (3) a *model structure* consisting of the relation of the model to other mathematical models for the specified process structure.

– **process-structure**

The process-structure is a logical expression over process and decomposition predicates that specifies the knowledge graph for which

<pre> model maximum-rate-MM variables input kp, E output Vmax formulation Vmax := kp*E end model </pre>	<pre> model saturation-MM variables input ks1, ks2, kp output Km formulation Km := (ks2+kp)/ks1 end model </pre>
<pre> model linear-MM variables input S, Vmax, Km output S, P operating conditions S > 2*Km formulation P := P(0) + S(0) - S S := integrate(-Vmax) end model </pre>	<pre> model exponential-MM variables input S, Vmax, Km output S, P operating conditions S < Km/2 formulation P := P(0) + S(0) - S S := integrate(-(Vmax/Km)*S) end model </pre>
<pre> model Michaelis-Menten variables input S, Vmax, Km output S, P formulation P := P(0) + S(0) - S S := integrate(-(Vmax*S)/(Km+S)) end model </pre>	

Figure 3. Mathematical models for enzymatic reactions assuming Michaelis-Menten kinetics.

the mathematical model is applicable. A term `process(pf)` specifies that the model describes the process frame `pf`. A term `decomposition(df)` specifies that the model describes the decomposition frame `df`. A conjunction of these terms specifies that the model describes a connected graph of the specified knowledge graph frames.

A negation term `not process(pf)` can be used to specify that the mathematical model is formulated under the assumption that the quantities in the knowledge graph are not influenced by the process frame `pf`.

Disjunctions of `process` or `decomposition` terms are not allowed, because an application frame specifies how a mathematical model can be applied to one knowledge graph. Consequently, when a mathematical model is applicable to different knowledge graphs, then separate application frames must be specified for each knowledge graph.

– interpretation

The interpretation specifies how variables of the mathematical model map to quantities in the qualitative level of the DESIMAL library. The

mapping need not be complete: variables of the mathematical model may represent quantities not included in the knowledge graph frames for which the model is applicable (e.g. $V_{\max}$ in the models displayed in Figure 3), and quantities in these knowledge graph frames may be left implicit in the model formulation. The interpretation is specified with the following terms:

A term `mapping(v,q)` specifies that the model variable v represents the quantity q . If this quantity does not occur in the knowledge graph frames for which the model is applicable, then v is an internal variable of the model.

A term `implicit(q)` specifies that the quantity q in the knowledge graph frames for which the model is applicable, is described implicitly in the formulation of the model. Thus, when this model is applied for the specified knowledge graph, then the quantity q is covered by this model.

– **model-structure**

The `model-structure` condition specifies relationships of a mathematical model with other mathematical models in the library. The `model-structure` condition is expressed as follows:

A term `req-model(m)` specifies that the current mathematical model can only be applied in combination with another mathematical model m . Examples are models that determine the parameters of other models.

A term `not model(m)` specifies another mathematical model m that cannot be used with the current mathematical model. These models are thus mutually exclusive. An example is an algebraic model and a model formulated with differential equations. Only one of these mutually exclusive mathematical models can be included in the simulation model.

4.3.1. *Example: the application frames*

Figure 4 shows the application frames for the models `linear-MM` and `maximum-rate-MM` in Figure 3. The first application frame specifies that the model `linear-MM` describes the behaviour of the quantity `Substrate` in the process frame `EnzymaticReaction`, and that the variables $V_{\max}$ and K_m represent quantities that are not included in the process frame.

The second application frame specifies that the variable $V_{\max}$ can be calculated by the model `maximum-rate-MM`. As this variable is used in the Michaelis-Menten equation only, the `model-structure` of this application frame specifies that the model `maximum-rate-MM` can only be used in combination with one of the Michaelis-Menten models in Figure 3.

These application frames show that a combination of models is needed to completely describe the process frame `EnzymaticReaction`.

```

application frame linear-MM
  process-structure
    process(EnzymaticReaction)
  interpretation
    mapping(S, Substrate)
    mapping(P, Product)
    mapping(Vmax, MM_MaximumRate)
    mapping(Km, MM_SaturationRate)
  end application frame

application frame maximum-rate-MM
  process-structure
    process(EnzymaticReaction)
  interpretation
    mapping(kp, EnzymaticReactionRate)
    mapping(E, ActiveEnzyme)
    mapping(Vmax, MM_MaximumRate)
  model-structure
    req-model(Michaelis-Menten) or
    req-model(linear-MM) or
    req-model(exponential-MM)
  end application frame

```

Figure 4. Application frames for two models for the process frame EnzymaticReaction.

5. The Automated Modelling Task in DESIMAL

Figure 5 gives an overview of the DESIMAL approach to automated construction of quantitative simulation models, called *Dynamic Product Models*, for postharvest physiological behaviour of agricultural products. The construction starts from a specification of the *phenomenon under study*. This specification consists of a set of quantities of interest, of which the behaviour over time must be explained, and a set of exogenous quantities, serving as the inputs of the simulation model.

The first subtask is called *Qualitative Process Analysis* (QPA), and constructs one or more *Process Structure Graphs* for the phenomenon under study. Each process structure graph describes an adequate decomposition of the phenomenon under study into the generic processes represented by the knowledge graph frames in the qualitative knowledge level of the DESIMAL library. Qualitative Process Analysis is described in Section 5.1.

Each process in the constructed process structure graphs corresponds to one or more models in the mathematical knowledge level. The second subtask is called *Simulation Model Construction* (SMC), and constructs a *Dynamic*

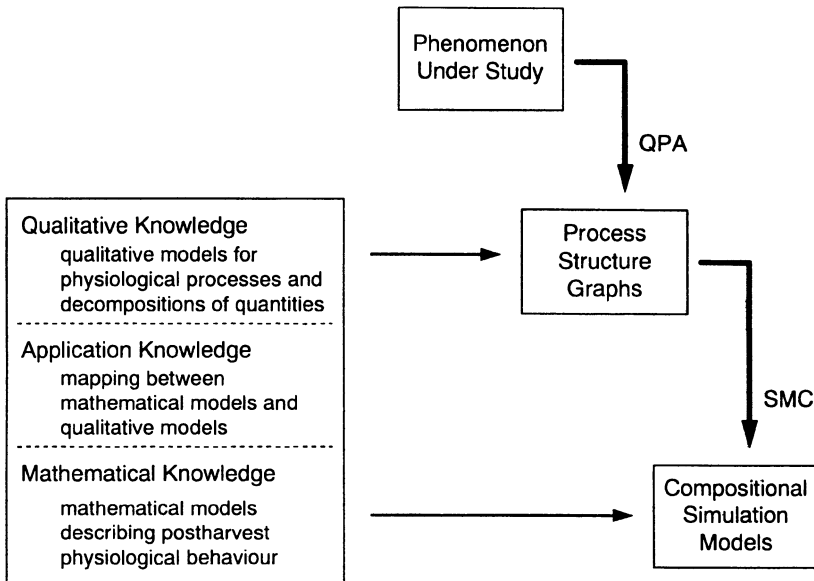


Figure 5. Overview of the DESIMAL method: the construction of a dynamic product model for a phenomenon under study is done in two steps. First, the phenomenon is decomposed into the (generic) processes in the qualitative knowledge level in the library, resulting in one or more process structure graphs. Second, dynamic product models are constructed using the mathematical models from the library as building blocks. The application knowledge is needed to connect the mathematical models to the processes.

Product Model for each process structure graph. A dynamic product model has a set of mathematical models and a simulation-control component that specifies when these models have to be used in the simulation. This subtask uses the application knowledge and the mathematical knowledge levels in the DESIMAL library. The mathematical knowledge level contains the mathematical models which are the building blocks of the dynamic product models. The application knowledge level provides the knowledge that is needed to select the appropriate mathematical models for the processes in the process structure graph. Simulation Model Construction is described in Section 5.2.

The construction of the process structure graphs and the construction of the dynamic product models are two separate modelling tasks, that both involve reasoning about the reusability of models in a model library. Component reuse in general involves three steps: retrieval, evaluation, and adaptation to fit new applications, see e.g. [25]. In the subtasks of the DESIMAL method, these steps have the following meanings:

- *Retrieval* involves the selection from the library of the building blocks that are applicable for the phenomenon under study.

- In QPA, this subtask retrieves all process frames that represent an (indirect) influence on a quantity of interest, and all decomposition frames that decompose a relevant quantity into more detailed subquantities.
- In SMC, this subtask retrieves all mathematical models that are applicable to one or more frames in the process structure graph developed for the phenomenon under study.
- *Evaluation* involves the restriction of the comprehensive models derived in the retrieval subtask to models that are adequate for the phenomenon under study.
 - In QPA, this subtask selects from the comprehensive process structure graphs those process frames and decomposition frames that represent an (indirect) influence of one of the exogenous quantities in the specification of the phenomenon under study on one of the specified quantities of interest.
 - In SMC, this subtask determines consistent model sets from the retrieved mathematical models. Each consistent model set is a simulation model that is adequate to describe the behaviour of the processes in a certain state of the simulation experiment.
- *Adaptation* of the submodels is not allowed. The knowledge graph frames and the mathematical models in the DESIMAL library are indivisible building blocks.

5.1. *Qualitative process analysis*

Each process structure graph constructed during Qualitative Process Analysis is a graph of interacting processes. Each process is an instance of a process frame retrieved from the qualitative knowledge level in the DESIMAL library. Two processes interact when a quantity that is affected by the one process, also affects the other process. As such a quantity belongs to the knowledge graph frames for the interacting processes, interactions need not be modelled explicitly. When different levels of detail are needed to decompose a phenomenon under study, a qualitative model contains instances of decomposition frames from the DESIMAL library, representing how an aggregate quantity is decomposed into subquantities. Below, the retrieval and evaluation subtasks of Qualitative Process Analysis are elaborated.

5.1.1. *Retrieval of relevant frames*

This retrieval subtask generates a set of candidate process structure graphs for the phenomenon under study. Each such process structure graph is a comprehensive description of the phenomenon under study at a certain level of detail, and contains all process frames and decomposition frames that

influence the quantities of interest at that level of detail. The candidate process structure graphs are constructed by the following steps:

- First, decomposition frames are retrieved for a quantity in the process structure graph. Each retrieved decomposition frame represents a change in the level of detail that may be needed to relate the quantity to the exogenous quantities. The process structure graph, as it has been constructed so far, is duplicated for each retrieved decomposition frame. The decomposition frame is only included in the duplicate process structure graph. The process structure graphs are further developed in parallel.
- Second, the process frames are retrieved that influence the quantity. Each retrieved process frame represents a separate influence on the quantity, that may be relevant for the phenomenon under study. These process frames are included in the current process structure graph.
- The frames that were added in the previous step may contain AKO and EQU relations that connect quantities in the frame with quantities in other knowledge graph frames in the DESIMAL library. For each group of AKO relations and for each group of EQU relations from one frame to a second frame, an instance of the second frame is created and added to the process structure graph.

The above steps are repeated until all decomposition frames and process frames have been found that are possibly relevant for the quantities in the candidate process structure graph that are not specified as exogenous.

5.1.2. *Selection of adequate process structure graphs*

The process structure graphs constructed in the retrieval subtask include all processes that influence the quantities of interest, and include all decompositions that may relate the quantities of interest to the exogenous quantities. The process structure graphs may therefore include frames, that are irrelevant for the phenomenon under study. In the evaluation subtask, each comprehensive process structure graph is simplified to an adequate process structure graph. A process structure graph is adequate for the phenomenon under study if the following constraints are satisfied:

- The process structure graph must contain all quantities of interest, but may leave out exogenous quantities that do not influence any of the quantities of interest.
- The process structure graph must contain all process frames that represent an (indirect) influence on a quantity of interest and that itself is (indirectly) influenced by an exogenous quantity.
- The process structure graph must contain all decomposition frames that introduce detailed quantities that are needed to relate a quantity of interest and an exogenous quantity.

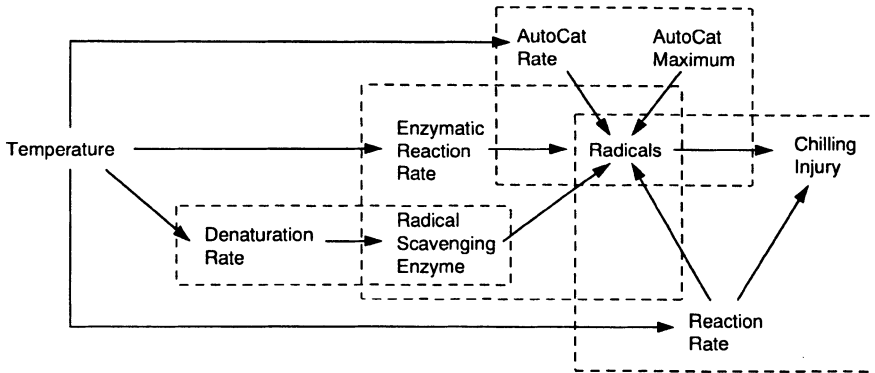


Figure 6. The process structure graph for chilling injury. The dashed boxes indicate the knowledge graph frames.

- The process structure graph must be a connected graph, to prevent the description of unrelated pieces of behaviour.

Hence, processes that cause changes in a quantity of interest, but that are themselves not affected by exogenous quantities in the phenomenon under study, are in principle treated as irrelevant.

The simplification of the comprehensive process structure graphs is done by a traversal over the FCAU, DCAU and PAR relations, starting from the exogenous quantities. This traversal finds the influence paths between the exogenous quantities and the quantities of interest. The frames that are not on these paths are considered to be irrelevant.

5.1.3. Example: process structure graph for chilling injury

Suppose for the chilling injury phenomenon, that ChillingInjury is the quantity of interest, and that Temperature is the exogenous quantity. As the chilling injury phenomenon can be modelled with process frames only (see Figure 2), Qualitative Process Analysis constructs one process structure graph.

The process structure graph is displayed in Figure 6, and is constructed as follows:

- First the process frame ChillingProcess is included, as this is the only process frame that contains the quantity of interest ChillingInjury.
- Next, the AKO relations from this process frame are followed, which causes instances of the process frames AutoCatalysis, ChemicalReaction, EnzymaticReaction and IrreversibleEnzymeDenaturation to be created and included in the process structure graph.
- The process frame ChemicalReaction only contains quantities and EQU relations to the process frames DegradationReaction and Formation-Reaction that specify the relations between these quantities. These

process frames are instantiated and included in the process structure graph.

- Finally, instances of the process frame *ReactionKinetics* are created for the quantities *ReactionRate*, *DenaturationRate*, and *Enzymatic-ReactionRate*. For the quantity *AutoCatRate* an instance of the process frame *ReactionSetKinetics* is created.

5.2. *Simulation model construction*

In the second subtask of DESIMAL, called *Simulation Model Construction*, separate quantitative simulation models, called *Dynamic Product Models* (DPM), are constructed for each of the process structure graphs that were constructed in Qualitative Process Analysis.

A dynamic product model consists of a set of mathematical submodels and a simulation-control component to activate the submodels. The simulation-control component reasons about the values of the variables of the submodels, about the occurrence conditions of the processes described by the submodels, and about the operating conditions of the submodels, to determine which of the submodels must be (de)activated. The simulation-control component is activated at each time point during a simulation experiment. In this way, a dynamic product model can describe the activation and deactivation of processes occurring in the phenomenon under study, and can describe the active processes with a variable set of active mathematical submodels.

Apart from the retrieval and evaluation tasks mentioned in Section 5, Simulation Model Construction involves an additional subtask to construct the simulation-control component of the dynamic product model. This subtask generates a declarative specification of the conditions under which the models in the dynamic product model have to be activated. The contents and the construction of the simulation-control component are discussed in Section 5.2.3. This additional task is needed only for Simulation Model Construction, because only the quantitative models are intended for simulation. Qualitative Process Analysis does not involve this subtask, because the qualitative models are not intended for simulation, but serve as requirements for Simulation Model Construction.

5.2.1. *Retrieval of applicable submodels*

In this subtask an *applicable model set* is retrieved from the mathematical models in the DESIMAL library. The applicable model set contains all models that describe a part of the behaviour represented in the process structure graph, and forms the library of the dynamic product model.

This subtask uses the process-structure conditions in the application frames in the DESIMAL library (see Section 4.3). A mathematical model is

applicable for a subgraph of the process structure graph, if the following conditions hold for the subgraph:

- the subgraph must contain all process frames and decomposition frames that are specified in the process-structure condition of the application frame for the mathematical model,
- the subgraph must be connected, and must not contain additional process frames or decomposition frames,
- the quantities in the subgraph must not be influenced by excluded process frames, as specified by not process terms in the process-structure condition of the application frame for the mathematical model.

These requirements express that all frames in the process structure graph are considered to be relevant, and therefore should be modelled. The subgraph may not have additional frames, as these are not considered in the selected mathematical model. The requirements are checked with a simple graph matching algorithm. The algorithm stops when applicable models have been found for all influence paths in the process structure graph, and when all models in the model library have been checked.

The exact mapping between the mathematical model and the subgraph in the process structure graph is specified in the interpretation of the application frame for the mathematical model. The interpretation determines which relations in the selected subgraph are covered by the formulation of the mathematical model, and thus for which relations other models have to be found. The interpretation also specifies which quantities in the subgraph are not represented by variables in the mathematical model, but are left implicit in the formulation of the mathematical model. Additional mathematical models must be retrieved for quantities in the subgraph of the process structure graph that are not represented by variables in the model, and that are not left implicit in the formulation of the model.

5.2.2. *Selection of consistent simulation models*

The *applicable model set* contains all mathematical models that describe one or more influence paths in the process structure graph. These models form the library of the dynamic product model. The simulation-control component in the dynamic product model then declaratively specifies which models have to be activated when.

An important requirement of the dynamic product model is that at each time point during a simulation experiment the set of active models forms a consistent simulation model for the processes that are occurring at that time point. The goal of the evaluation subtask of Simulation Model Construction is to select such consistent simulation models from the applicable model set.

A consistent simulation model is a set of mathematical models for the process structure graph, that satisfies the following conditions:

- Each relation in the process structure graph is modelled by exactly one mathematical model.
- The simulation model includes all models that are required by req-model terms in the model-structure condition in the application frames for the mathematical models in the simulation model.

The req-model terms for a mathematical model specify other models that have to be included in the simulation model before the mathematical model can be applied. For example, the submodels maximum-rate-MM and saturation-rate-MM are only applicable if the model Michaelis-Menten or one of its approximation models are included in the simulation model.

If a relation in the process structure graph can be described by several applicable models, then for each applicable model, a separate simulation model is constructed. For example, three models are applicable for the relations in the process frame *EnzymaticReaction: Michaelis-Menten*, *linear-MM*, and *exponential-MM*. Hence, in this case three simulation models are constructed, one for each applicable model for the process frame.

The dynamic product model must describe the behaviour of the phenomenon under study under one set of underlying assumptions. Which models are mutually exclusive is specified in the model-structure conditions of the application frames: a term *not model (m1)* in the application frame for a mathematical model *m2* specifies that the models *m1* and *m2* are mutually exclusive.

The dynamic product model cannot contain mutually exclusive models. Hence, the set of consistent simulation models must be restricted so that the models all assume a common set of underlying assumptions.

The models *Michaelis-Menten*, *linear-MM* and *exponential-MM* are not mutually exclusive, so that these models can all be included in the dynamic product model. A model that assumes first-order kinetics for the reaction is mutually exclusive with *Michaelis-Menten* kinetics, and thus cannot be included in the dynamic product model.

5.2.3. *Construction of the simulation-control component*

The third subtask of the Simulation Model Construction task constructs the simulation-control component of a dynamic product model. The simulation-control component is activated for each time point during a simulation experiment, and specifies the conditions under which the submodels in a dynamic product model must be activated and deactivated. A submodel is active at a certain time point, if (1) the process described by the submodel is occurring at that time point, (2) the values of the variables of the submodel are all

known, and (3) the operating conditions of the submodel are satisfied. In the simulation-control component, these conditions are checked separately at each time point during a simulation run.

A simulation-control component consists of three sections: Initialisation, Execution, and Termination.

The Initialisation section specifies the initial values for variables in the dynamic product model, and may also activate submodels to calculate variables that depend only on variables that have a constant value throughout the simulation experiment. These submodels are thus activated only once, and must be time-independent. The Initialisation section is only evaluated at the start of the dynamic product model.

The Execution section is a rule base that is evaluated at each time point during a simulation experiment to determine which submodels must be activated or deactivated to simulate the behaviour at the current time point. The section contains three groups of rules:

- **Identification of occurring processes**

As processes need not be active throughout a simulation experiment, the first step is to determine which processes are actually occurring. Only models for these processes need to be activated.

Which processes are occurring is determined from the occurrence conditions of the processes. The occurrence condition of a frame is stored in the qualitative knowledge level of the DESIMAL library, and specifies under which conditions the behaviour represented by the frame is occurring.

- **Selection of applicable models**

The second group of rules are the applicable-model rules. The simulation-control component has an applicable-model rule for each mathematical model in the dynamic product model. Each applicable-model rule specifies for which (combinations of) occurring processes the model is applicable, and specifies the operating conditions of the submodel. The specification of the processes for which the model is applicable is retrieved from the application frame of the submodel.

- **Selection of the active submodels**

Finally, the applicable submodels are selected that must be activated to simulate the behaviour at the current time point.

The simulation-control component has an activate-model rule for each submodel in the dynamic product model. The condition part of an activate-model rule specifies which values must be calculated before the model can be activated. This ensures that the models are activated in the order of computation.

If several submodels are applicable for one process, then the activate-model rules must also specify which model is to be activated.

Finally, the Termination section contains a terminate-simulation condition that specifies when the dynamic product model normally has to terminate. The dynamic product model terminates successfully, if this condition is satisfied. The dynamic product model terminates unsuccessfully if at a certain time point no submodels can be activated. In that case, the simulation has reached a state that was not foreseen during model construction, so the dynamic product model was used outside its region of operation.

5.2.4. *Example: the radical scavenging reaction*

The radical scavenging process in the chilling injury phenomenon is an enzymatic process. The DESIMAL library contains several models for enzymatic reactions, displayed in Figure 3. These models are included in the dynamic product model to describe the radical scavenging process in the chilling injury phenomenon. Figure 7 displays the rules in the Execution section of the simulation-control component that concern the models.

The first rule specifies that the radical scavenging process (which is a kind of EnzymaticReaction) occurs as long as there are free Radicals and as long as the RadicalScavengingEnzyme is not completely denatured.

The next five rules specify which models are applicable for this process. The models linear-MM and exponential-MM are only applicable under the specified operating conditions. The other models are always applicable whenever the radical scavenging process is occurring.

The last five rules specify when these models can be activated. The models maximum-rate-MM and saturation-rate-MM must be activated first, as these models calculate the new values for $V_{\max}$ and K_m , respectively. The last rule in Figure 7 specifies that the model Michaelis-Menten must only be applied when the approximation models are not applicable.

6. Discussion

We have presented the DESIMAL method for automated construction of quantitative simulation models for postharvest physiological behaviour of agricultural products.

The DESIMAL method uses a library that consists of two separate description levels. These levels are connected by an intermediate level containing explicit knowledge about the applicability of the mathematical models for the knowledge graphs in the qualitative knowledge level. This separation between physiological processes and mathematical models enables the

```

if Radicals > 0 and RadicalScavengingEnzyme > 0
then occurring-process( EnzymaticReaction )

if occurring-process( EnzymaticReaction )
then applicable-model( Michaelis-Menten )
if occurring-process( EnzymaticReaction )
  and Radicals > 2*Km
then applicable-model( linear-MM )
if occurring-process( EnzymaticReaction )
  and Radicals < Km/2
then applicable-model( exponential-MM )
if occurring-process( EnzymaticReaction )
then applicable-model( maximum-rate-MM )
if occurring-process( EnzymaticReaction )
then applicable-model( saturation-rate-MM )

if applicable-model( maximum-rate-MM )
then activate-model( maximum-rate-MM )
if applicable-model( saturation-rate-MM )
then activate-model( saturation-rate-MM )
if applicable-model( linear-MM )
  and calculated( Km ) and calculated( Vmax )
then activate-model( linear-MM )
if applicable-model( exponential-MM )
  and calculated( Km ) and calculated( Vmax )
then activate-model( exponential-MM )
if applicable-model( Michaelis-Menten )
  and calculated( Km ) and calculated( Vmax )
  and not applicable-model( linear-MM )
  and not applicable-model( exponential-MM )
then activate-model( exponential-MM )

```

Figure 7. The Execution section in a simulation-control component of a dynamic product model for the radical scavenging process in the phenomenon of chilling injury.

modeller to decompose the phenomenon under study into a set of physiological processes, without the need to consider the mathematical details of these processes. As in the modelling support approaches reviewed in Section 2.1, the qualitative knowledge level in the DESIMAL library is an intermediate level between the modeller's knowledge about a phenomenon under study, and the mathematical models describing physiological behaviour. In the automated

modelling approaches reviewed in Section 2.2, where these description levels are much more interwoven.

The knowledge graph formalism used to represent the qualitative knowledge is based on [22, 23]. An important difference between these formalisms lies in the use of the frames. In the original knowledge graph formalism, the frames are concepts in the graph and thus can be related to other concepts. In DESIMAL, however, relationships between frames, such as generalisation of processes and decomposition of processes, are represented by EQU and AKO relations between the quantities of the frames:

A generalisation relationship between two processes is in our approach modelled by AKO relations between the quantities in the two process frames, to explicitly represent which quantities in the generic frame are instantiated for quantities in the specific frame. For example, the AKO relations in Figure 2 between the quantities in the process frames ChillingInjury and ChemicalReaction represent that the occurrence of chilling injury is modelled as an chemical reaction, in which Radicals are consumed to produce ChillingInjury. When using AKO relations at the level of frames the instantiation of the quantities cannot be explicitly represented anymore.

A decomposition of a process into subprocesses is modelled by introducing intermediate quantities that are influenced by the subprocesses. The quantity that is affected by the complex process is then a composition of the intermediate quantities. For example, vessel blocking in flower stems is complex process that may hamper the uptake of water through the flower stem, thereby leading to flower wilting. Vessel blocking can be caused by bacteria and by air bubbles entering the flower stem. The decomposition of this complex process is represented at the level of quantities (see Figure 8): the decomposition frame Dec-BlockedVessels specifies that the quantity BlockedVessels is decomposed into the quantities BacteriaBlockedVessels and AirBlockedVessels, which are affected by the process frames BacteriaBlocking and AirBlocking, respectively.

This is in contrast with Rickel's method [14, 15], in which two constructs are used to represent differences in level of detail. On the one hand, *explanation* relations are used to represent that a process can in more detail be described as a set of subprocesses with intermediate quantities.² On the other hand, *encapsulation* relations are used to represent that a quantity can be decomposed into subquantities with intermediate processes. In our view, the explanation and encapsulation relations represent the same kind of knowledge (namely about possible decompositions), but from different viewpoints (explanations of the processes and encapsulations of the quantities). The decomposition frames in DESIMAL approach correspond to encapsulation relations.

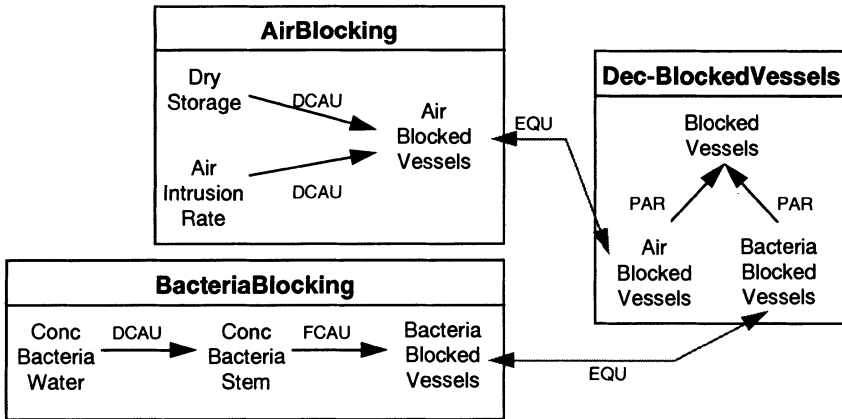


Figure 8. Knowledge graph representing that the complex process of vessel blocking is decomposed at the level of quantities.

Rickel's method constructs qualitative models for answering prediction questions in the domain of plant physiology. The building blocks of these models are the individual causal relations between domain quantities rather than sets of causal relations grouped into processes. Apart from the usual qualitative representation of direction of change (+, -), in Rickel's method a distinction is made between functional influences, in which the one quantity is a function of the other, and differential influences, in which the rate of change of one quantity is a function of another quantity. This distinction is also used in the DESIMAL library.

To determine which part of the physiological knowledge is relevant for the analysis, in Rickel's method information is used about the time scales at which changes in quantities occur. Functional influences are time-independent, but each differential influence has an explicit time scale at which the change occurs that is represented by the influence. Only those influences are considered to be relevant that have a time scale equal or smaller than the time scale of interest. Thus, if the quantities in the question change at a time scale of minutes, then the changes that take hours or days are considered to be negligible.

This approach works only for short-term qualitative analysis of physiological phenomena under well-defined initial conditions. However, the simulation models constructed by the DESIMAL method are applied in analysis of the behaviour of agricultural products during long-term distribution chains. Therefore, time-scale abstraction is in our case invalid, because the simulation period will be much longer than the time-scale at which the quantities change. For example, if the specification of the phenomenon under study only con-

tains quantities that change at the scale of minutes and the simulation model is used to analyse the behaviour during a period of five days, changes with a scale of hours and days will be important as well. These changes would be left out in Rickel's method.

The Simulation Model Construction task is similar to the approach of Iwasaki and Levy [12], since in both approaches simulation models are constructed that during a simulation experiment select the active model set from the submodels that are applicable for the system under study. The selection algorithm in our approach is explicitly specified in the simulation-control component of the simulation model, whereas in the approach Iwasaki and Levy the active models are determined from the specifications of the model fragments in the simulation model.

Acknowledgements

The author wishes to thank Jan Treur and Mark Willems and the anonymous reviewers for their extensive comments on earlier versions of this article.

Notes

¹ This is a simplified version of the example used in [12].

² Actually, the explanation relation is defined on the influences, but an influence can be seen as a small process.

References

1. Sloof, M., Tijskens, L. M. M. & Wilkinson, E. C. (May 1986). Concepts for Modelling Quality of Perishable Products. *Trends in Food Science and Technology* 7: 165–171.
2. Sloof, M. & Simons, A. E. (1993). Knowledge Based Construction of Models to Predict Cut Flower Quality. In *Modelling and Simulation, Proceedings of the European Simulation Multiconference, ESM'93*, 13–17. Lyon, France, June 7–9.
3. Sloof, M. & Simons, A. E. (1994). Towards a Task Model for the Design of Simulation Models. In *Proceedings of the Modelling and Simulation Conference, ESM'94*, 721–725, Barcelona, Spain, June 1–3.
4. Murray, K. & Sheppard, S. (March 1988). Knowledge-Based Simulation Model Specification. *Simulation* 50(3): 112–119.
5. Top, J. L. (1993). *Conceptual Modelling of Physical Systems*. PhD dissertation, University of Twente, Enschede, September.
6. Mili, F. (1995). User Oriented Library Documentation. In *Proceedings of First International Workshop on Knowledge-Based Systems for the (Re)Use of Program Libraries, KBUP'95*, 11–20. Sophia-Antipolis, France, November 23–24.
7. Top, J. L., Breunese, A. P. J., Broenink, J. F. & Akkermans, J. M. (1995). Structure and Use of a Library for Physical Systems Models. In *Proceedings ICBGM'95*, 97–102, Las Vegas.

8. Forbus, K. D. (1990). Qualitative Physics: Past, Present, and Future. In Weld, D. S. & de Kleer, J. (eds.) *Readings in Qualitative Reasoning about Physical Systems*, 11–39. Morgan Kaufmann Publishers, Inc.
9. Weld, D. S. & de Kleer, J. (eds.) (1990). *Readings in Qualitative Reasoning about Physical Systems*. Morgan Kaufmann Publishers, Inc.
10. Falkenhainer, B. & Forbus, K. D. (1991). Compositional Modeling: Finding the Right Model for the Job. *Artificial Intelligence* 51: 95–143.
11. Falkenhainer, B. & Forbus, K. D. (1992). Composing Task-Specific Models. In DSC-Vol. 41, *Automated Modeling*, ASME 1992, 1–9.
12. Iwasaki, Y. & Levy, A. Y. (1994). Automated Model Selection for Simulation. In *Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI-94*. Seattle, Washington, July 31–August 4.
13. Pandurang Nayak, P. (1995). *Automated Modeling of Physical Systems*. Lecture Notes in Computer Science, vol. 1003. Springer.
14. Rickel, J. & Porter, B. (1994). Automated Modeling for Answering Prediction Questions: Selecting the Time Scale and System Boundary. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, 1191–1198. Seattle, Washington, July 31–August 4.
15. Rickel, J. (1995). *Automated Modeling of Complex Systems to Answer Prediction Questions*. PhD thesis, University of Texas at Austin.
16. Schut, C. & Bredeweg, B. (1996). An Overview of Approaches to Qualitative Model Construction. *The Knowledge Engineering Review* 11(1): 1–25.
17. Xia, S. & Smith, N. (1996). Automated Modelling: A Discussion and Review. *The Knowledge Engineering Review* 11(2): 137–160.
18. Gruber, T. R. (1993). Model Formulation as a Problem-Solving Task: Computer-Assisted Engineering Modeling. *International Journal of Intelligent Systems* 8(3): 105–127. Special volume on Knowledge acquisition as modeling.
19. Addanki, S., Cremonini, R. & Penberthy, J. S. (1991). Graphs of Models. *Artificial Intelligence* 51: 145–177.
20. Tijskens, L. M. M., Otma, E. C. & van Kooten, O. (1994). Photosystem II Quantum Yield as a Measure of Radical Scavengers in Chilling Injury in Cucumber Fruits and Bell Peppers: A Static, Dynamic and Statistical Model. *Planta* 194: 478–486.
21. Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Reading, MA: Addison-Wesley.
22. James, P. (1991). Structuring Knowledge using Knowledge Graphs. In *Proceedings of the 5th European Knowledge Acquisition for Knowledge-based Systems Workshop (EKAW 91)*. Strathclyde University, Scotland.
23. Willems, M. (1993). *Chemistry of Language: A Graph-Theoretic Study of Linguistic Semantics*. PhD dissertation, University of Twente, Enschede, January.
24. Forbus, K. D. (1984). Qualitative Process Theory. *Artificial Intelligence* 24: 85–168.
25. Penix, J. & Alexander, P. (1995). Design Representation for Automating Software Component Reuse. In *Proceedings of First International Workshop on Knowledge-Based Systems for the (Re)Use of Program Libraries, KBUP'95*, 75–84. Sophia-Antipolis, France, November 23–24.

Address for correspondence: M. Sloof, Everest, Reitscheweg 55, 5232 BX 's-Hertogenbosch, The Netherlands. Telephone: +31.73.6450460; Fax: +31.73.6450920

A Neuro-Fuzzy Approach to Identify Lettuce Growth and Greenhouse Climate

B.T. TIEN and G. VAN STRATEN

*Systems and Control Group, Department of Agricultural Engineering and Physics,
Wageningen Agricultural University, Bomenweg 4, 6703 HD Wageningen, The Netherlands
(Tel.: +31-(0)317-484953; Fax: +31-(0)317-484819; E-mail: ufo@hermes.com.tw)*

Abstract. A hybrid neuro-fuzzy approach called the NUFZY system, which embeds fuzzy reasoning into a triple-layered network structure, has been developed to identify nonlinear systems. A set of membership functions at the input layer is partially linked with a layer of rules, using pre-set parameters. By means of a simplified centroid of gravity defuzzification method, the output becomes linear in the weights. Therefore, very fast estimation of the weight parameters can be achieved by using the orthogonal least squares (OLS) method, which also provides a method to efficiently remove the redundant fuzzy rules from the prototype rule base of the NUFZY system. In this paper, the NUFZY system is applied to identify lettuce growth and greenhouse temperature from real experimental data.

Results show that the NUFZY model with the fast OLS training can perform quite well in predicting both lettuce growth and greenhouse temperature. In contrast to the mechanistic modeling procedures, the neuro-fuzzy approach offers an easier route and a fast way to build the nonlinear mapping of inputs and outputs. In addition, the resulting internal network structure of the NUFZY system is a self-explanatory representation of fuzzy rules. Under this frame, it is a perspective that one is able to incorporate the human knowledge in this approach, and, hopefully, to deduce any interpretable rules that describe the systems' behavior.

Key words: neuro-fuzzy modeling, orthogonal least squares; fuzzy rule reduction, plant growth, greenhouse climate

Introduction

In horticultural crop production, a major goal of greenhouse operation is to create the most suitable environment for crops cultivated inside such that the best economic returns can be obtained at harvest. By changing the indoor conditions in the greenhouse the quality and quantity of the crop can be improved. In order to achieve this improvement, in addition to the grower's long term accumulated practical experience, computer controlled operation of the greenhouse is indispensable. In recent times, the optimal control of greenhouse climate has been studied [1]. However, the application of optimal control of greenhouse climate relies strongly on the availability of models that describe the dynamics of indoor climate and plant growth as a function of the control variables. As such an optimal control approach, lots of work

has to be done on experiments for model validation and parameter calibration. The procedure to establish such a mechanistic model is usually time-consuming and costly. In situations where the experimental data are available, a data-driven modeling or black-box modeling approach, for example, neural networks modeling and fuzzy modeling, can be an alternative attractive to cumbersome mechanistic modeling.

Neural networks are characterized by performing nonlinear mapping from the space of independent variables to the space of dependent variables by parallel architectures consisting of simple processing elements that communicate through weighted connections. They are able to approximate functions or to solve certain control problems by learning from examples via parameter optimization [2]. In contrast, fuzzy modeling can be regarded as grey box modeling which is capable of processing available expert knowledge or experience in the form of linguistic IF/THEN – fuzzy rules and membership functions. From the identification point of view, just like neural networks, also fuzzy systems are universal approximators that can approximate any real continuous function on a compact set to arbitrary accuracy [3, 4]. Applications of fuzzy modeling of complex nonlinear systems have been reported [5, 6]. Meanwhile, based on the structural similarity and functional equivalence between fuzzy systems and neural networks, several fuzzy-neuro or neuro-fuzzy structures have been proposed [7–10]. The integrated fuzzy-neuro system seeks to combine the advantages of both paradigms and concurrently compensates for their weaknesses. In our early work, a preliminary study on the integrated neuro-fuzzy system, named the NUFZY system, has been done and applied to identify a tomato growth process [11, 12].

The goal of this paper is to develop a technique, based on the neuro-fuzzy scheme, that can perform nonlinear modeling of multivariable systems, like the crop growth model and the greenhouse climate model. Compared to the regular mechanistic modeling procedure, this neuro-fuzzy approach offers an easier route and a fast way to build the nonlinear mapping of inputs and outputs, as it makes use of the available experimental data. In contrast to pure neural networks, the neuro-fuzzy approach forms a fuzzy-rule-like internal connection, which is a self-explanatory representation of fuzzy rules. This particular structure is attractive because it gives the perspective that one is able to incorporate the human knowledge in this approach, such as intuitions on the operational fuzzy rules and the applied membership functions, and, hopefully, to deduce any interpretable rules that describe the systems' behavior.

With respect to the horticultural crop production, for control purposes the crop growth models under development are preferred to be dynamic models because they enable the prediction of crop performance as related to the particular status of the crop and the environmental conditions at any given

moment [13]. It is also required to develop dynamic models of the greenhouse climate. It is noticed that crop growth and indoor climate are interrelated to each other and essentially can be regarded as a multivariable process. By taking these crop states and climate condition as inputs to the fuzzy modeling, it can be expected that a large number of fuzzy rules are needed when the NUFZY system is applied to approximate such a multivariable process. In order to avoid overfitting the process, the redundancy in this large fuzzy rule base must be taken into account. Hence, the identification problem concerns, here, both the determination of the model structure, as well as the estimation of the parameters. The structure problem in the present neuro-fuzzy approach is partly related to the required number of fuzzy rules. By means of the orthogonal least squares (OLS) method, it is shown that the parameters can be estimated and the redundant fuzzy rules in the prototype fuzzy rule base can be detected at once.

This paper is organized as follows. The structure of the developed NUFZY system and its corresponding characteristics are presented in section 2. Section 3 presents the orthogonal least squares identification and fuzzy rule reduction for the NUFZY system. Examples are demonstrated in section 4 and the results are discussed in section 5.

2. Structure of NUFZY

The developed NUFZY system, as shown in Figure 1, is characterized by a triple-layered feedforward network.

The first and second layer of the NUFZY system deal with the antecedent part of the fuzzy rule base and the third layer concerns the consequent part. The NUFZY system performs a Takagi and Sugeno type of fuzzy inference [5], i.e., the consequent part is formed as a linear combination of the premise variables. A special case is obtained while the outputs are not linear functions but taken as crisp real values (constant terms of the linear functions), which is also the method used in the NUFZY system. Given a system with ni input variables x_i , $i = 1, \dots, ni$, and with no outputs $\hat{y}_j$, $j = 1, \dots, no$, the fuzzy rules can be expressed in the form

$$\begin{aligned} R^r: & \text{ IF } (x_1 \text{ is } A_{k1}^r(x_1) \text{ AND } \dots \text{ AND } x_i \text{ is } A_{ki}^r(x_i) \text{ AND } \dots \\ & \text{ AND } x_{ni} \text{ is } A_{kni}^r(x_{ni})) \text{ THEN } (\hat{y}_1^r = g_1^r, \dots, \hat{y}_j^r = g_j^r, \hat{y}_{no}^r = g_{no}^r) \quad (1) \end{aligned}$$

where superscript r denotes the r^{th} fuzzy rule and $A_{ki}^r(x_i)$, defined by a bell-shaped membership function, represents the ki^{th} linguistic label of x_i with respect to the fuzzy rule R^r , and the membership function in the consequent part is expressed as a singleton value, denoted by g_j^r . The AND connection

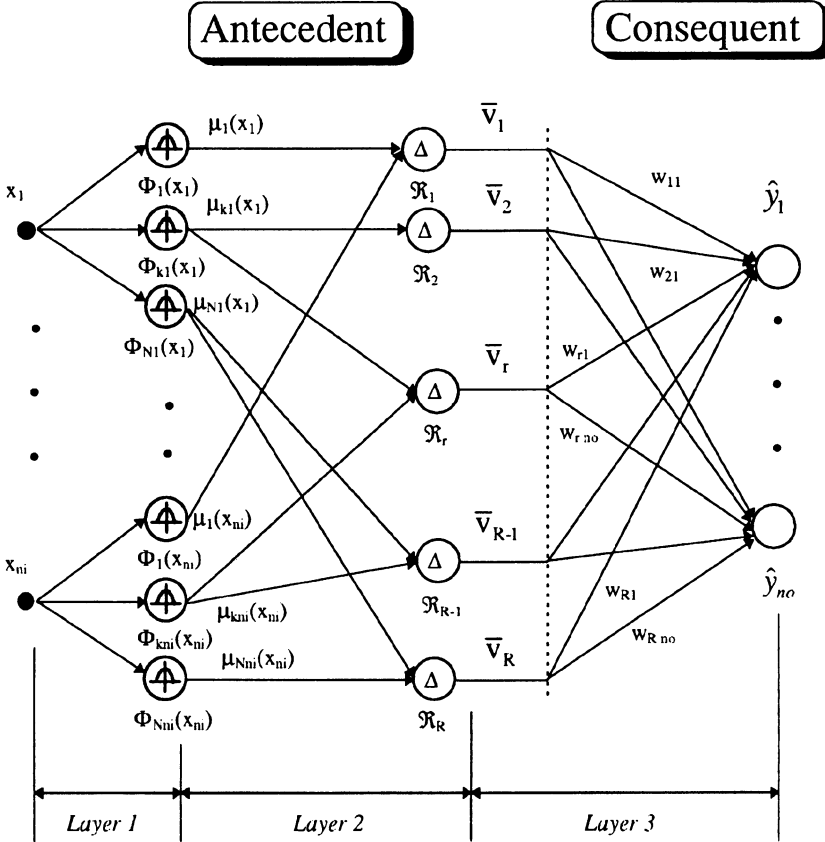


Figure 1. The structure of the NUFZY system.

in Eq. (1) is implemented as the algebraic product. After aggregating the independent contribution of each fuzzy rule, a simplified centroid of gravity (COG) defuzzification is used to generate a crisp result. In the following an outline is given of the essential parts of the NUFZY system.

Layer 1

The input node x_i , just serves to distribute the input into the first layer nodes with fixed weights of unity. The node on the first layer, called *membership node* and denoted as $\Phi_{ki}(x_i)$, represents a membership function that performs fuzzification of the input variables. Each x_i has its own N_i linguistic labels characterized by the membership functions $\Phi_{ki}(x_i)$ that are numbered as 1, 2, ..., ki , ..., Ni , for $i = 1, \dots, ni$. Inputs x_i 's are used directly as arguments of

the membership functions and to yield the corresponding graded membership values. Hence, the fuzzified values $\mu_{ki}(x_i)$ according to the following inverse multiquadratic IMQ membership function (abbreviated as IMQ)¹ is given by,

$$\mu_{ki}(x_i) = [(x_i - c_{i,ki})^2 + \sigma_{i,ki}^2]^{-\frac{1}{2}} \quad \text{with } ki = 1, \dots, Ni, i = 1, \dots, ni \quad (2)$$

where $c_{i,ki}$ and $\sigma_{i,ki}$ are the ki^{th} center and bandwidth of $\Phi_{ki}(x_i)$, respectively. For ease of notation, the $\mu_{ki}(x_i)$ are reordered sequentially and denoted as $\alpha_m(x_i)$ with subscript $m = 1, \dots, M$, where M is the total number of membership function of all input variables. M can be obtained by $M = N_1 + \dots + N_{ni} = \sum N_i, i = 1, \dots, ni$, and $\alpha_m(x_i) = \mu_{ki}(x_i)$ with index m ,

$$m = \sum_{j=1}^{i-1} N_j + ki \quad \text{with } i = 1, \dots, ni. \quad (3)$$

Hence, the center c_m and bandwidth σ_m corresponding to $c_{i,ki}$ and $\sigma_{i,ki}$ are defined accordingly. For a specific input vector $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_i \ \dots \ x_{ni}]^T$, the corresponding graded membership values can be denoted as $\boldsymbol{\alpha} = [\alpha_1 \ \dots \ \alpha_m \ \dots \ \alpha_M]^T$ which just stacks the outputs of the membership functions for all inputs. In the practical application, the initial values of these node parameters, $c_{i,ki}$ and $\sigma_{i,ki}$ (or c_m and σ_m), can be determined by the designers, for instance, from the available experimental data or their own experience and knowledge.

Layer 2

In this layer the node, called a *rule node*, represents a fuzzy rule and is denoted as $\mathfrak{R}_r, r = 1, 2, \dots, R$; where R is the total number of all fuzzy rules given by $R = \prod N_i, i = 1, \dots, ni$. Inputs to the *rule node* are the graded membership values from the *membership node* of layer one. Each *rule node* performs a two-step operation as will be described later. According to Eq. (1) each *rule node* involves only one membership function for each input. Therefore, the existence of a connection between a *rule node* and a *membership node* is represented with a value of either 1 or 0, forming a relationship matrix $\mathbf{RM}$ with dimension $(R \times M)$, where M is the total number of all membership functions of all inputs as described above. Each row r of $\mathbf{RM}$, represents the status of the antecedent part of a fuzzy rule, i.e., a value of 1 represents a link between the r^{th} *rule node* and the corresponding *membership node*, whilst an element with value 0 indicates no connection. The use of the $\mathbf{RM}$ matrix can facilitate the calculation of fuzzy reasoning. The following example clarifies the content of the $\mathbf{RM}$ matrix. Suppose there are three inputs, x_1, x_2, x_3 , and each has 2, 3, 2 membership functions with values $\mu_{11}(x_1), \mu_{12}(x_1), \mu_{21}(x_2), \mu_{22}(x_2), \mu_{23}(x_2), \mu_{31}(x_3), \mu_{32}(x_3)$, respectively (or, corresponds to $\alpha_1(x_1)$,

$\alpha_2(x_1), \alpha_3(x_2), \alpha_4(x_2), \alpha_5(x_2), \alpha_6(x_3), \alpha_7(x_3)$, respectively). In this case, $R = 2 \times 3 \times 2 = 12$ and $M = 2 + 3 + 2 = 7$. A relationship matrix denoted as $RM_{(2,3,2)}$ then is a 12 by 7 matrix as follows.

Relationship Matrix $RM_{(2,3,2)}$							
m =	1	2	3	4	5	6	7 (=M)
	μ_{11}	μ_{12}	μ_{21}	μ_{22}	μ_{23}	μ_{31}	μ_{32}
r = 1	1	0	1	0	0	1	0
r = 2	0	1	0	1	0	1	0
r = 3	1	0	0	0	1	1	0
r = 4	0	1	1	0	0	1	0
r = 5	1	0	0	1	0	1	0
r = 6	0	1	0	0	1	1	0
r = 7	1	0	1	0	0	0	1
r = 8	0	1	0	1	0	0	1
r = 9	1	0	0	0	1	0	1
r = 10	0	1	1	0	0	0	1
r = 11	1	0	0	1	0	0	1
r = 12=R	0	1	0	0	1	0	1

Hence, the relationship matrix constructs a prototype fuzzy rule base with all possible combination of input variables provided each N_i is assigned. Form the above example, it can be seen that the third fuzzy rule ($r = 3$) is composed by $\alpha_1(x_1)$, $\alpha_5(x_2)$, and $\alpha_6(x_3)$ (they correspond to membership values $\mu_{11}(x_1)$, $\mu_{23}(x_2)$, and $\mu_{31}(x_3)$, respectively). Obviously, when the number of input variables increases, this prototype fuzzy rule base will become large exponentially.

There is a two-step operation to generate the node output in this layer. First, the algebraic product is used to realize the “AND” connection in Eq. (1), so that a transient firing strength of each rule, v_r , is obtained as a function of inputs x_i ’s via the membership function α_m (or $\mu_{i,ki}$)

$$v_r = \Pi_{m=1}^M RM(r, m) \alpha_m = RM(r, a_i : b_i) \mu_i \quad (4)$$

where Π is a product operation taking into account the connections defined by the RM matrix. $RM(r, a_i : b_i)$ represents the partial elements from a_i to b_i of the r^{th} row vector of the RM matrix, where $b_i = \Sigma N_k$, $k = 1, \dots, i$, and $a_i = b_i - N_i + 1$. Vector μ_i is given by $\mu_i = [\mu_{i1} \ \mu_{i2} \ \dots \ \mu_{iki} \ \dots \ \mu_{iNi}]^T$. Eq. (4) specifies that v_r is obtained by just multiplying the membership functions involved in rule r , according to the connections shown in Figure 1. Second, to normalize v_r the normalized firing strength $\bar{v}_r$ is calculated as

$$\bar{v}_r = \frac{v_r}{\sum_{r=1}^R v_r} \quad (5)$$

This normalized firing strength $\bar{v}_r$ represents the output of the *rule node*.

Layer 3

The node, denoted as $\hat{y}_j$, stands for the NUFZY system output. The link in this layer represents a weight parameter denoted as w_{rj} , for $r = 1, \dots, R, j = 1, \dots, no$, and connects node $\hat{y}_j$ and $\mathfrak{R}_r$. The parameter w_{rj} actually represents the unknown variable g_j^r in the consequent part of the fuzzy rule given in Eq. (1). By applying the simplified centroid of gravity (COG) defuzzification method this node then performs a summation operation such that the model output is obtained by

$$\bar{y}_j = \frac{\sum_{r=1}^R v_r \times w_{rj}}{\sum_{r=1}^R v_r} = \sum_{r=1}^R w_{rj} \bar{v}_r = \mathbf{w}_j^T \bar{\mathbf{v}} \quad (6)$$

where $\mathbf{w}_j$ is the weight parameter vector given by $\mathbf{w}_j = [w_{1j} \dots w_{rj} \dots w_{Rj}]^T$ and $\bar{\mathbf{v}}$ is a normalized firing strength vector given by $\bar{\mathbf{v}} = [\bar{v}_1 \dots \bar{v}_r \dots \bar{v}_R]^T$ with element $\bar{v}_r$ defined by Eq. (5). It is noted that the model output is linear in the weight parameters. This becomes a preferred situation for identification since the property of linear-in-the-parameter allows the solution of the unknown weight parameters to be identified by some standard least squares parameter estimation methods, as an example, the orthogonal least squares method (shown later). Besides, in the NUFZY system the linear property will be retained for other choices of *t-norm* operators for the **AND** connection in the fuzzy rules as well as for other choices of the membership functions, such as Gaussian, triangular, and trapezoidal shaped membership function. Also note that the $\bar{v}_r$ can be viewed as a fuzzy basis function (FBF), so that the NUFZY output forms a FBF expansion [3]. Thus, the NUFZY system has the property of a universal function approximator.

It is interesting to compare the relation of the NUFZY system to the radial basis function networks (RBF)[14]. The main difference is that characteristic information from the system is embedded in the fuzzy-rule-like internal connections of the NUFZY system. This may allow the NUFZY system to represent the system behavior in a rule form deduced from the data set. Further, in the NUFZY system the antecedent parts (layer 1 and 2) are only partially connected (defined by the **RM** matrix) rather than the full connection in the RBF networks. However, the partial connection in the NUFZY network does not affect its function as a universal approximator.

3. Least Squares Identification

Least squares (LS) identification is an effective optimization tool that yields a unique solution for the values of the parameters in a linear regression model. It can be seen from Eq. (6) that the output of the NUFZY system forms as a linear combination of weight parameters. Hence, one may employ the LS method to identify these weight parameters in Eq. (6).

Given a set of training data with np events, the multi-input-multi-output NUFZY system can be expressed as a linear regression model in matrix form as

$$Y = \bar{V}W + E \quad (7)$$

where Y is the $np \times no$ desired output, $\bar{V}$ is the $np \times R$ normalized firing strength matrix which follows from Eq. (4) and Eq. (5). W is the $R \times no$ weight matrix, and E is a $np \times no$ matrix of model errors. The estimate parameter $\hat{W}$ satisfies the following normal equation

$$\bar{V}^T \bar{V} \hat{W} = \bar{V}^T Y \quad (8)$$

and can be solved in a least squares sense just by taking the pseudo-inverse of the normal equation of Eq. (8). I.e., the parameter $\hat{W}$ is then obtained by

$$\hat{W} = (\bar{V}^T \bar{V})^{-1} \bar{V}^T Y \quad (9)$$

However, this ordinary least squares method suffers from the singular value problem when $\bar{V}^T \bar{V}$ is ill-conditioned or not invertible. In that case $\hat{W}$ will be seriously effected by round-off errors accumulating during the calculation. In order to avoid the numerical problem, the orthogonal least squares (OLS) method based on the classical Gram-Schmidt method is a better alternative of ordinary least squares computation [14]. In addition, this method offers a way to reduce the fuzzy rule base by deleting redundant fuzzy rules to achieve a moderate model size. The OLS method is briefly explained below.

3.1. Orthogonal least squares method

In the OLS method matrix $\bar{V}$ is decomposed into QA such that (7) becomes

$$Y = (QA)W + E = QG + E \quad \text{with } G = AW \quad (10)$$

where $Q = [q_1 \dots q_r \dots q_R]$ is an $np \times R$ matrix with orthogonal column vectors q_r , $r = 1, \dots, R$, and A is a $R \times R$ invertible upper triangular matrix. Therefore, the least squares solution of Eq. (10) is given by

$$\hat{G} = (Q^T Q)^{-1} Q^T Y \quad (11)$$

Once the matrix $\hat{G}$ is obtained, the orthogonal least squares solution $\hat{W}$ is then given by

$$\hat{W} = A^{-1}\hat{G} \quad (12)$$

Since matrix A is upper triangular, the inverse of A is very easy to achieve by backward substitution in an iterative manner. The orthogonal decomposition of $\tilde{V}$ into QA based on the classical Gram-Schmidt method is described in [14].

3.2. Fuzzy rule selection and deletion

By increasing the number of fuzzy rules or regressors one may achieve a better approximation to the unknown system. However, too many fuzzy rules makes a large rule base which becomes difficult to interpret. Moreover, there is always the danger of overfitting, where the final result is heavily determined by the noise in the data. Hence, a procedure is required to find out the redundant fuzzy rules that do not make contribution. Provided that the prototype fuzzy rule base is determined already, the main theme of applying the OLS method is to select the moderate fuzzy rules such that a set of R_s significant rules, that make the largest contribution to the variance of the desired output Y in (7), are selected from the initial R rule base, while $R_s \leq R$.

During the OLS procedure, the values of an “error reduction ratio”, $[err]$, of each candidate regressor, i.e., each fuzzy rule, will be calculated on every orthogonalization step and only the one which has the maximum $[err]$ value is selected and added to the model on that step. In case the model contains certain regressors being highly correlated, which means that they are almost linearly dependent, it will result in an ill-conditioned problem. To detect this condition, one can just check the norm of the newly added orthogonalized regressor to the model: $\|q_r\|^2$. A value of $\|q_r\|^2 = q_r^T q_r = 0$, simply implies that the newly selected regressor q_r is a linear combination of formerly selected orthogonalized regressors in the model, and thus does not add to the information content of the model and therefore should be kicked out from the model. The norm will practically never become exactly equal to zero. Therefore, in this paper, a small value of 10^{-7} is specified to be a threshold value of the norm $\|q_r\|^2$ in order to check the linear dependence of orthogonalized regressors.

The orthogonalization procedure terminates when all the candidate fuzzy rules have been processed. Among the results is a sequence of indices ranking the significant rules, and a list of indices of fuzzy rules which have been removed. Consequently, the total number of rules at the end, R_s , is less than the theoretical number of rules belonging to the given input partition, provided almost linear dependent fuzzy rules have indeed occurred. A new

set of weights is then calculated according to the selected linear independent rules that are used as a final rule base and the weights belonging to those almost linear dependent rules are assigned to be zeros.

3.3. *Implementation remarks*

As shown in section 2, a prototype of the fuzzy rule base can be constructed whenever the number of membership functions of each input variable, N_i , is chosen. In practice, N_i as well as the parameters in layer 1 (center, c , and bandwidth, σ) can be determined by the designers. Once a set of experimental data is on hand, the parameter values of c and σ can, alternatively, be estimated from the data. So that N_i becomes the only parameter that has to be assigned. In this paper we adopt the latter approach since it allows an easy way to deal with the initialization of parameter values. In the following examples in section 4, the values of N_i are found by systematically varying N_i from 2 up to 4 or 5 and selecting the values that gives the best performance. Once N_i is specified, the centers of membership function are chosen as equally spaced on the range of x_i and the value of three times standard deviation of x_i is taken as its bandwidth. During the identification procedure, layer 1 parameters, c and σ , are kept constant for every specific prototype fuzzy rule base and the OLS method is only used to estimate the weight parameter, w , of layer 3 and to detect the redundant fuzzy rules. Hence, the determination of the most effective structure and the estimation of the optimal parameters (in the least squares sense) can be carried out at once by the OLS method. In summary, in addition to estimate the weight parameters, the OLS algorithm mainly acts as a tool to eliminate the redundant or insignificant fuzzy rules from the prototype NUFZY system.

4. Examples and Simulation Results

In the optimal control of greenhouse crop production, one has to deal with different time scales: the slow crop growth and the fast greenhouse physics. In the work of [1], a two-time scale decomposition using a singular perturbation method was employed to solve the optimal control problem. The same idea is used in this paper to separate the crop growth problem and the greenhouse physics problem. For crop growth, the fast dynamics of climatic conditions are regarded as irrelevant and one can utilize the mean climatic values for the crop growing period. On the other hand, during the short term identification of the greenhouse climate the states of the crop are assumed to be constant.

From the point of view of control applications the models should have predictive ability. Accordingly, in the two examples in this section, a NUFZY

model is established to predict the lettuce growth and the greenhouse temperature. In the development of models, first a set of data gathered from previous experiments is taken as training set such that the output weight parameters (w in Eq. (6)) of the NUFZY model are identified by the above OLS method. With these identified parameters, another independent set of data, named validation set, is used to evaluate the prediction ability of the identified NUFZY model.

4.1. Lettuce growth identification

Problem description

In [1], a lettuce growth model is described by a single state variable, namely total dry weight X_d . The governing differential equation is

$$\frac{dX_d}{dt} = c_\beta(c_\alpha\phi_{\text{phot}} - \phi_{\text{resp}}) \quad (13)$$

where c_β and c_α are conversion parameters; ϕ_{phot} and ϕ_{resp} , represent gross photosynthesis gain and maintenance respiration loss of lettuce, respectively. They are complex nonlinear functions of the dry weight itself and some input variables, Z_c (greenhouse indoor CO_2 concentration), Z_t (greenhouse indoor air temperature) and V_i (outdoor radiation). The parameters in these relationships were determined empirically. Further details can be found in [1]. Two experiments were done to calibrate the model parameters and to validate the above model. The lettuce used in the experiments is *Lactuca sativa* L., which was grown in an experimental greenhouse in Wageningen from 17/10/1991–16/12/1991 (cultivar “Berlo”) and from 21/1/1992–17/3/1992 (cultivar “Norden”). The greenhouse was under computer control according to the rules followed in normal Dutch horticultural practice. During the two experiments, destructive measurements of the lettuce (10 and 9 sampling dates, respectively) were performed, whereas the greenhouse climate, the actuators and the outdoor climate conditions were recorded for further analysis.

Eq. (13) defines a continuous-time model of the lettuce growth rate, which is a relationship between the biomass and the external input variables. In the set up of the NUFZY model, the same variables are used. First, the time step is taken as one day (24 hours), using the daily averaged measurements of V_i , Z_t and Z_c as approximations of the input signals. Due to the fact that the lettuce dry weight is not available every day for these experiments, linearly interpolated values are taken as estimates of the data. Hence, a modified discrete-time model, based on daily averaged data of V_i , Z_t and Z_c and linear interpolated X_d , is applicable to approximate the continuous growth of lettuce. It is represented as

$$\frac{\Delta X_d(k)}{\Delta T(k)} = f_1(X_d(k), Z_c(k), Z_t(k), V_i(k)) = \frac{X_d(k+1) - X_d(k)}{T(k+1) - T(k)} \quad (14)$$

where k is the discrete time index, $X_d(k)$ is the dry weight of lettuce at day $T(k)$; $f_1(\cdot)$ represents the nonlinear function to be identified. Therefore, a one-day-ahead prediction of lettuce dry weight according to Eq. (14) can be obtained as

$$X_d(k+1) = X_d(k) + f_1(X_d(k), Z_c(k), Z_t(k), V_i(k)) \cdot \Delta T(k) \quad (15)$$

An alternative formulation of a one-day-ahead prediction of lettuce dry weight is to incorporate the previous values of inputs in the nonlinear function directly,

$$X_d(k+1) = f_2(X_d(k), Z_c(k), Z_t(k), V_i(k)) \quad (16)$$

NUFZY model establishment

In [1], a sensitivity analysis has been done on the lettuce growth model of Eq. (13) suggesting that dry matter production of lettuce is mainly dominated by a limited number of inputs. Especially, the CO_2 concentration has a stronger positive effect on dry matter production than greenhouse indoor air temperature. Besides, it can be seen that the indoor air temperature strongly correlates to the outdoor radiation. Hence, on establishing the NUFZY model, the function variables have been restricted to outdoor radiation V_i , indoor CO_2 concentration Z_c , and the present state of crop X_d . Further, a derived quantity temperature-day is commonly used in practice, which suggests that the summation of temperature values over the growing periods should be taken into account. In terms of outdoor radiation V_i , this temperature-day quantity can be replaced by another quantity, accumulated radiation, $\sum V_i(k)$ (denoted as $ACV_i(k)$), which sums up the daily averaged radiation over time $T(k)$. Therefore, for the NUFZY modeling the chosen variables are X_d , Z_c and ACV_i . Formally, the goal of the identification of the lettuce growth is to establish the NUFZY models, f_{NUFZY1} and f_{NUFZY2} , such that they can approximate the unknown nonlinear functions f_1 and f_2 in Eq. (15) and Eq. (16), respectively. Here, f_{NUFZY1} and f_{NUFZY2} are called the first kind and the second kind of NUFZY modeling. It can be seen that in the first kind of NUFZY modeling the one-step-ahead prediction of lettuce growth is obtained indirectly based on the inferred growth rate, whilst the second kind of NUFZY modeling infers the one-step-ahead prediction of lettuce growth directly. The predicted dry weight of lettuce by the NUFZY model, $\hat{X}_d(k+1)$ can be written as: The first kind of NUFZY modeling

$$\hat{X}_d(k+1) = X_d(k) + f_{\text{NUFZY1}}(X_d(k), Z_c(k), ACV_i(k)) \cdot \Delta T(k) \quad (17)$$

and the second kind of NUFZY modeling

$$\hat{X}_d(k+1) = f_{\text{NUFZY2}}(X_d(k), Z_c(k), ACV_i(k)) \quad (18)$$

When using the model in a real application, the biomass is usually not observed. Therefore, the validation was done in such a way that no measurements of X_d were employed except for the initial point. Thus, on validation of the NUFZY model, Eq. (17) and Eq. (18) become

$$\hat{X}_d(k+1) = \hat{X}_d(k) + f_{\text{NUFZY1}}(\hat{X}_d(k), Z_c(k), ACV_i(k)) \cdot \Delta T(k) \quad (19)$$

and

$$\hat{X}_d(k+1) = f_{\text{NUFZY2}}(\hat{X}_d(k), Z_c(k), ACV_i(k)) \quad (20)$$

where the input variable $\hat{X}_d(k)$ is obtained from the prediction of the trained NUFZY model at previous step $k - 1$. At the next step the present estimate is then iteratively fed into NUFZY model as input set. Note, this method is called parallel method [2] and the validation by this method is therefore a n-step ahead. In contrast, another way of validation, the so called serial-parallel method can also be employed, like Eq. (17) and Eq. (18), which just takes the real value of X_d into the NUFZY model, provided the measurements of X_d are available. The training and validation process of NUFZY modeling is given as follows.

Training process

The training of the NUFZY model is done with data taken from experiment 1 (17/10/1991–16/12/1991, in total 61 days). In this set of training data, the values of daily averaged CO_2 concentration Z_c (ppm) and accumulated daily averaged outdoor radiation ACV_i (W/m^2) as well as linear interpolated dry weight of lettuce X_d (g) are treated as external inputs. Hence, in total, 60 tuples $[X_d(k), Z_c(k), ACV_i(k), \Delta X_d(k)/\Delta T(k)]$ or $[X_d(k), Z_c(k), ACV_i(k), X_d(k+1)]$ are used to train the first and second kind of NUFZY modeling, respectively. From the training set, when each N_i is assigned (see below), the centers of the IMQ membership functions (c in Eq. (2)) are taken equally spaced in each input range whilst to the width, σ , of membership functions of each input three times the value of the standard deviation is assigned. The weight parameter set θ_1 or θ_2 (w_j in Eq. (6)) of the NUFZY model is estimated by the OLS method such that the sum of square errors $\sum_k [\Delta X_d(k)/\Delta T(k) - f_{\text{NUFZY1}}(\cdot; \theta_1)]^2$ or $\sum_k [X_d(k+1) - f_{\text{NUFZY2}}(\cdot; \theta_2)]^2$ is minimized, where $k = 1$ to 60. The number of membership functions for each input, N_i , ($i = 1, 2$, and 3, corresponding to X_d , Z_c , and ACV_i , respectively) is varied from 2 to 4. Therefore, the initial number of fuzzy rules ranges from $2 \times 2 \times 2 = 8$ to $4 \times 4 \times 4 = 64$. By taking the sum of squared errors as a criterion function, the final structure used is the one that has the lowest sum of squared errors.

Validation process

After obtaining the weight parameter set θ_1 or θ_2 , another set of data taken from experiment 2 (21/1/1992–17/3/1992, in total 56 days) is used to validate the first and second kind of NUFZY models. Due to the values of $X_d(k)$ is unknown at every moment except the sample dates, the validation process is carried out according to the parallel method. The mean value of the measured X_d on the first sample date, 0.15 g, is used as an initial value $X_d(1)$. Values of the other inputs $Z_c(k)$ and $ACV_i(k)$ are measurable for the whole experiment period. So, in the first step, the estimate is obtained from Eq. (17) or Eq. (18) with available measurements $Z_c(k-1)$ and $ACV_i(k-1)$ (using the parameter set θ_1 or θ_2), whereas in the following sequence, this estimate is iteratively fed into Eq. (19) or Eq. (20) together with $Z_c(k)$ and $ACV_i(k)$, in order to generate the next step prediction of.

Results

Figure 2 and Figure 3 demonstrate the results of both models for the training and the validation process. The results of training process show that the first kind of NUFZY modeling, denoted as NUFZY1($3 \times 4 \times 3:15; \text{IMQ}$), can achieve a nice approximation, where the notation NUFZY1($3 \times 4 \times 3:15; \text{IMQ}$) indicates that 3, 4, and 3 IMQ membership functions are assigned to X_d , Z_c , and ACV_i , respectively. As a result of the OLS identification, only 15 fuzzy rules are chosen to be involved in the fuzzy rule base and, consequently, 21 redundant rules have been removed from the initial 36 ($= 3 \times 4 \times 3$) rules. The best result of the second kind of NUFZY modeling is NUFZY2($2 \times 2 \times 3:10; \text{IMQ}$), so that X_d , Z_c , and ACV_i have 2, 2 and 3 IMQ membership functions, respectively, and the number of identified fuzzy rules is only 10. For the purpose of comparison, in these figures, the mean values of the dry weight measurements with 95% confidence intervals are plotted as vertical bars and the simulated results from [1], by Eq. (13), are presented as well. It can be seen that both results obtained from $f_{\text{NUFZY1}}(\cdot; \theta_1)$ and $f_{\text{NUFZY2}}(\cdot; \theta_2)$ perform as well as the mechanistic model Eq. (13) during the training process and are slightly less accurate to the result of the mechanistic model in the validation process. However, the number of estimated parameters (15 and 10, respectively) in the NUFZY models compares favorably to the 19 parameters of the mechanistic model. One example (Rule 6) of the generated fuzzy rules of the first kind of NUFZY modeling is listed below, which suggests that when the crop is in the beginning stage of development (X_d is small, A_1 , and ACV_i is low, C_1), the very high CO_2 concentration (Z_c is B_4) will result in a negative effect on the growth rate. Other identified rules are alike to it and will not be presented here.

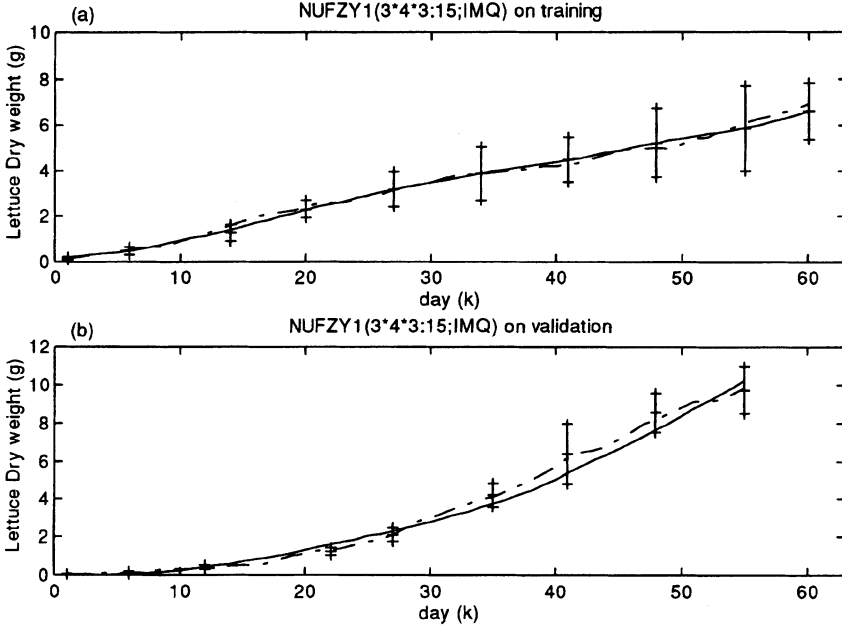


Figure 2. Prediction of lettuce dry weight by NUFZY1($3 \times 4 \times 3; 15; \text{IMQ}$) during experiment 1, training (one-step-ahead) (a); and experiment 2, validation (seasonal prediction) (b). Solid line indicates output from NUFZY1 and dash-dot line indicates result from the one state variable model, Eq. (13), described in [1]. The vertical bars show a 95% confidence interval around the mean value of the measurements.

Rule 6: **IF** $X_d(k)$ is A_1 **AND** $Z_c(k)$ is B_4 **AND** $ACV_i(k)$ is C_1
THEN $\Delta X_d(k)/\Delta T(k) = -2.0663$

where the centers and widths of linguistic variables $[A_1 \ A_2 \ A_3]$ of input $X_d(k)$ (g) are $[0.15 \ 3.38 \ 6.61]$ and $[1.9532 \ 1.9532 \ 1.9532]$, respectively. For $Z_c(k)$ (ppm) with the linguistic variables $[B_1 \ B_2 \ B_3 \ B_4]$, centers and widths are $[422.65 \ 498.05 \ 573.46 \ 648.86]$ and $[61.10 \ 61.10 \ 61.10 \ 61.10]$, respectively; whereas for the linguistic variables $[C_1 \ C_2 \ C_3]$ of $ACV_i(k)$ (W/m^2), $[21.62 \ 420.01 \ 818.41]$ and $[210.47 \ 210.47 \ 210.47]$, respectively.

4.2. Greenhouse temperature identification

In contrast to the slow response of the crop growth, the greenhouse climate shows very distinct fast dynamics. Based on a time scale in minutes, the effect of crop growth is assumed to be negligible. In the following example the

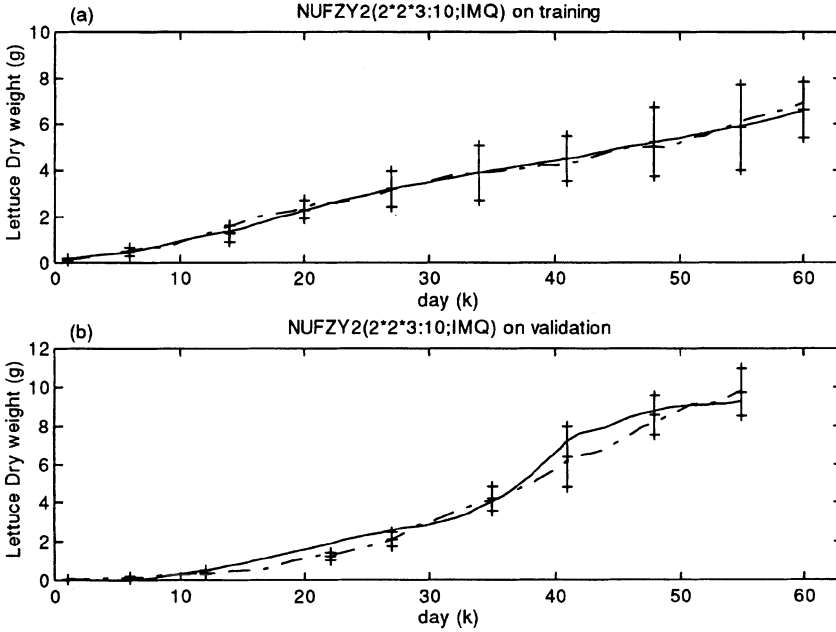


Figure 3. Prediction of lettuce dry weight by NUFZY2(2×2×3:10;IMQ) during experiment 1, training (one-step-ahead) (a); and experiment 2, validation (seasonal prediction) (b). Solid line indicates output from NUFZY2 and dash-dot line indicates result from the one state variable model, Eq. (13), described in [1]. The vertical bars shows a 95% confidence interval around the mean value of the measurements.

NUFZY system demonstrates the modeling of the dynamics of the greenhouse temperature and performs a one-step-ahead prediction based on the present indoor states, control inputs, and outdoor disturbances. Extensions to other climatic states is possible but will not be discussed here.

Problem description

For control purposes, several dynamic models of greenhouse physics have been developed, such as [15] and modified model as [16]. In [17], based on energy and mass balances, the dynamics of the greenhouse temperature Z_t is described by a first order differential equation with heat losses from natural ventilation, through the roof cover and to the soil; and heat gains from the heating pipes and from solar radiation. The differential equation is written as

$$\frac{dZ_t}{dt} = \frac{1}{c_g} [(k_v + k_r)(V_t - Z_t) + k_s(Z_s - Z_t) + k_p(Z_p - Z_t) + \eta V_i] \quad (21)$$

where k_v , k_r , k_s , and k_p are the effective heat transfer coefficients of ventilation, roof cover, soil and heating pipe, respectively. V_t , Z_s , and Z_p stand

for outdoor air temperature, indoor greenhouse soil temperature and heating pipe temperature, respectively. η and V_i are radiation efficiency factor and disturbance from outdoor radiation, respectively. In Eq. (21) the control of window opening U_w and two disturbances, outdoor wind speed V_s and wind direction V_d , are implicitly involved in the coefficient k_v ; whereas control of the heat supply U_t determines the heat pipe temperature Z_p . Hence, taking these implicit relations into account Eq. (21) can be written formally as

$$Z_t(k+1) = f_3(Z_t(k), Z_s(k), U_w(k), U_t(k), V_t(k), V_s(k), V_d(k), V_i(k)) \quad (22)$$

where $f_3(\cdot)$ is a complex nonlinear function describing the greenhouse temperature dynamics as a function of the input variables described above.

NUFZY model establishment and results

In this example, greenhouse climate data are taken from the experimental results of [18]. The experiment is done on tomato production where one greenhouse compartment is controlled by a receding horizon optimal control (RHOC) algorithm in order to compare it to another compartment controlled by a current commercial greenhouse climate control computer. The experiment was conducted from 1/8/1995 till 30/10/1995. All the input variables mentioned in Eq. (22) were measured every minute. In the following, the training and validation data used for the NUFZY modeling makes use of those measurements and records taken from the optimal controlled greenhouse compartment.

From Eq. (22) it is seen that the large number of input variables causes a large fuzzy rule base of the NUFZY model. For ease of modeling, the higher order effects induced by the state of soil Z_s are neglected. It is also observed that the windward and lee side windows are almost fully open all day long in August because of the high air temperature inside the greenhouse. Hence, in this demonstration, the input variables, U_w , V_s , and V_d , can be further left out by properly choosing those days on which the windows are open. Therefore, a NUFZY model with restricted validity is established to give a one-step-ahead prediction of Z_t with reduced input variables as

$$\hat{Z}_t(k+1) = f_{\text{NUFZY3}}(Z_t(k), U_t(k), V_t(k), V_i(k)) \quad (23)$$

where the input variables are greenhouse temperature Z_t ($^{\circ}\text{C}$), heating valve opening U_t (0–100%), outdoor air temperature V_t ($^{\circ}\text{C}$) and radiation V_i (W/m^2). The effect of the disregarded input variables will appear as unmodeled effects in the model errors.

A set of data originating from 23/8/1995–25/8/1995 is taken as training data. During this period, the windows are almost always fully open. In order to reduce the amount of data to be processed, measurements based on every five

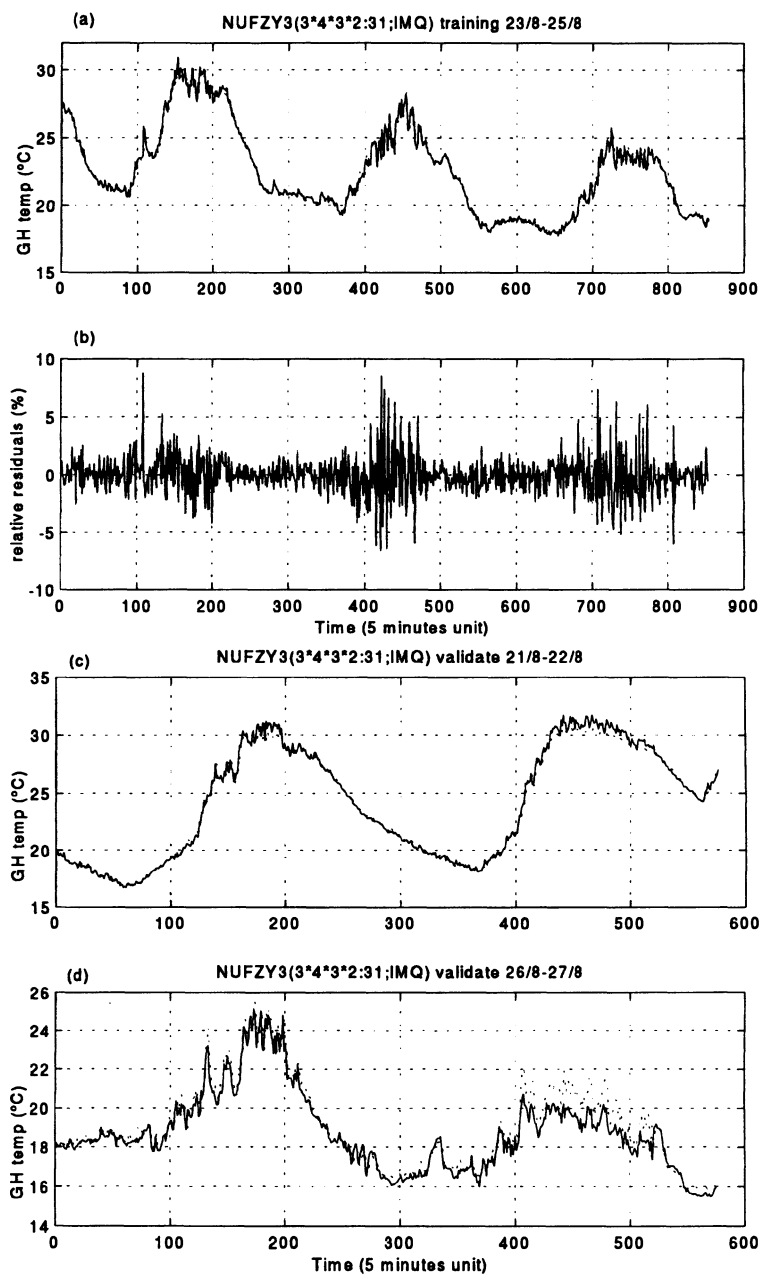


Figure 4. Simulated one-step-ahead prediction of greenhouse temperature by NUFZY3 ($3 \times 4 \times 3 \times 2:31;IMQ$) during 23/8–25/8, training process (a) and relative residuals of training (b); validation on date 21/8–22/8 (c); 26/8–27/8 (d); 31/8–4/9 (e); 21/9–25/9 (f) and 16/10–20/10 (g). Solid line: measured greenhouse temperature; dotted line: NUFZY prediction. In figure (a) and (c), it is hardly to recognize the difference between the measurement and the NUFZY prediction.

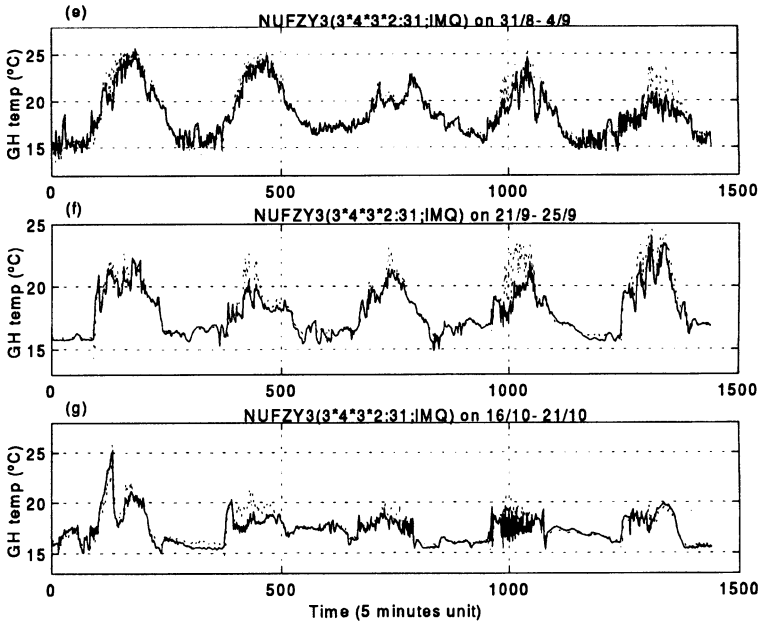


Figure 4. Continued.

minutes are taken for analysis. The training process, similar to the previous example, is proceeded by the OLS method. Independent data sets on several dates in three different periods of the experiment, *viz.* in the beginning (21/8–22/8, 26/8–27/8, and 31/8–4/9), in the middle (21/9–25/9), and toward the end (16/10–20/10), are taken to validate the identified NUFZY model. In contrast to the previous example, only the one-step-ahead prediction capability was tested, using the measured temperature as the input for the next prediction (serial-parallel prediction). In agriculture, such methods could be used for control.

Results of the training and validation process are shown in Figure 4. The identified model is NUFZY3(3×4×3×2:31;IMQ), showing that, to input variables Z_t , U_t , V_t , and V_i are assigned 3, 4, 3, and 2 IMQ membership functions, respectively, and the final number of identified fuzzy rules is 31. For the training data process, Figure 4(a) shows that a fairly good fit is obtained with a reduced set of fuzzy rules, since 41 rules are removed from the prototype rule base ($3 \times 4 \times 3 \times 2 = 72$). Figure 4(b) illustrates the relative error of the residuals is less than $\pm 10\%$ (around ± 2 °C). The larger discrepancies mainly occurs when the outdoor radiation disturbance has large fluctuations, *i.e.*, during day time. The validation results in the beginning period (Figure 4(c) date 21/8–22/8, (d) 26/8–27/8, and (e) 31/8–4/9) possess

a fair fit to the measured data. However, in the other two periods large discrepancies can be found. This result confirms that black box models are firmly data dependent and need retraining as the process slowly moves to other operating regions.

5. Discussion

In the example of lettuce growth, two kinds of NUFZY models, f_{NUZFY1} and f_{NUZFY2} , are used. Results show that both of them have comparative prediction ability as that of Eq. (13), a mechanistic model describing the lettuce growth. Yet, different prediction behavior can be found from them. It is noticed from Figure 2(b) that the increment model f_{NUZFY1} leads to a smoother growth curve than the direct model f_{NUZFY2} in Figure 3(b). However, it should be kept in mind that if the sampling interval becomes large (i.e., less sampling is done), the linear interpolation used for training of both NUFZY models should be circumvented since it leads to a model that tries to mimic a piecewise linear growth curve rather than the true growth curve of lettuce. In this example, however, the sampling interval is around one week, which does not cause severe problems in this respect. Another factor of interest is that although the models were trained to yield one-day-ahead predictions, the validation was done by running the model for the whole growing season of experiment 2 without updating from the actual measurements (a parallel method). In this approach, there exists a risk that the prediction becomes worse because of the accumulation of model errors. In this particular example the results remain within the measurement uncertainty, thus indicating that, from the training data set, the NUFZY model does match up the lettuce growth dynamics sufficiently well to achieve a reasonable seasonal prediction, which could be used, e.g., in the production scheduling.

In the example of identifying the greenhouse temperature, it is the fast fluctuation (within the 5 minutes period) of outdoor radiation during day time that results in larger model errors. Also, by neglecting the effects of window opening and wind speed, the NUFZY model occasionally tends to overestimate the greenhouse temperature. For instance on 27/8 in Figure 4(d) and on 24/9 in Figure 4(f), it can be checked that the wind speed increased from the range 0–6 m/s (used in the training process) to 0–10 m/s. This creates an extra heat flux due to extra ventilation out of the real greenhouse which, of course, does not occur in the present model. As the identification is done by off-line training, this phenomenon confirms, again, that the NUFZY model is data dependent – just like any other black box approach – and that the model can only be expected to perform well if it is used under similar conditions as encompassed in the training process. In order to contrast a more

general model for long term prediction, further study is needed to involve more climate factors into the NUFZY model. Or, alternatively, for control purposes, a recursive identification can be used which adjusts the model to the most recent data, for example, in [12] a recursive adaptation of the parameters of the NUFZY system has been studied, showing the feasibility for on-line application.

One of the most appealing promises of applying a fuzzy system is that the grower/expert's knowledge can be utilized and incorporated into the framework of the fuzzy system. However, in the present paper this feature has not been used. The only merit gained from the present NUFZY approach is that it just uses available experimental data to construct a specific model to carry out the approximation function, like an ordinary neural network does. Yet, Compared to the mechanistic models such as Eq. (13) and Eq. (21), the NUFZY modeling saves a lot of work on parameter calibration and model development, provided a comparable model accuracy is required.

In respect to the use of expert's knowledge in the fuzzy modeling, there are several possibilities to incorporate qualitative information into fuzzy models. For example, by collecting the expert's knowledge and then directly aggregating it as fuzzy rules which are suitable for fuzzy modeling. In practice, this approach is not easy to implement since usually further refinement of those aggregated fuzzy rules is needed to match the modeling task. More specifically, parameters of membership functions of each fuzzy rule have to be defined by trial and error. Another approach of incorporating qualitative information is to take it as an initial setting of parameters in both the antecedent and the consequent part of fuzzy rules [19]. During the training process, the parameters are trained in order to give a good match of the model to the given data. As a result the final rules may be quite different from the original rules proposed by the experts. This suggests a contradiction on the consistency of qualitative information used in such an adaptive fuzzy system. If the qualitative information used in the initialization does contain the key behavior of the unknown system, this method will just facilitate the convergence of the fuzzy system; otherwise, the contradiction remains. Another aspect that seems to interfere with the ability to insert qualitative information is the observation that if the antecedent part of the fuzzy rule is fixed, like in the NUFZY model, still a good model accuracy can be achieved by merely tuning the consequent weights of the fuzzy rules [12]. It is, however, conceivable that simultaneous training of the antecedent and the consequent parameters would allow models with much less rules which may be easier to interpret. In any case, more work is needed in order to clarify the issue of utilizing qualitative information.

6. Conclusions

A hybrid neuro-fuzzy system called the NUFZY system was developed to identify nonlinear systems. It is characterized by a triple-layered network consisting of a membership layer, a rule base layer, and a linear output layer. Once the number of membership functions of each input variable has been determined, a prototype fuzzy rule base is then constructed as a partially connected network by means of a relationship matrix. By using the simplified centroid of gravity defuzzification method, the output is linear in the weights, and the weight parameters can be determined by an orthogonal least squares algorithm based on the classical Gram-Schmidt decomposition. Moreover, this procedure opens a convenient way to remove linearly dependent or almost linearly dependent fuzzy rules from the prototype rule base, thus avoiding redundancy in the rule base.

In this paper, the NUFZY system is used to identify a developing system – lettuce growth –, and a system with a stationary operating point – greenhouse temperature – from real experimental data. In the case of the lettuce growth, the established NUFZY model offers seasonal prediction as a function of the solar radiation and CO₂ concentration as well as the state of lettuce itself. In contrast, the greenhouse temperature model was evaluated as a one-step-ahead prediction model, suitable for control application in the greenhouse production. Results presented show that the NUFZY model with the fast OLS training algorithm and a reduced fuzzy rule base give a suitable identification of the lettuce growth and greenhouse temperature.

Acknowledgment

The authors are grateful to Dr. E. J. van Henten and Ir. R. F. Tap for kindly offering the experimental data.

Note

¹ Other choices of the membership function, such as Gaussian, triangular, and trapezoidal, are also possible. In this paper we confine ourselves just on discussion of IMQ membership function. In the particular examples of [12], similar performance can be found by implementing the Gaussian type of membership function.

References

1. van Henten, E. J. (1994). *Greenhouse Climate Management: An Optimal Control Approach*. PhD thesis, Wageningen Agricultural University, Wageningen, The Netherlands.

2. Narendra, K. S. & Parthasarathy, K. (1990). Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Trans. Neural Networks* 1: 4–27.
3. Wang, L. X. & Mendel, J. M. (1992). Fuzzy Basis Function, Universal Approximation, and Orthogonal Least-Squares Learning. *IEEE Trans. Neural Networks* 3: 807–814.
4. Castro, J. L. (1995). Fuzzy Controller Are Universal Approximators. *IEEE Trans. Syst., Man, Cybern* 25: 629–635.
5. Takagi, T. & Sugeno, M. (1985). Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Trans. Syst., Man, Cybern* 15: 116–132.
6. Sugeno, M. & Tanaka, K. (1991). Successive Identification of a Fuzzy Model and Its Applications to Prediction of a Complex System. *Fuzzy Sets and Systems* 42: 315–334.
7. Lin, C. T. & Lee, C. S. G. (1991). Neural-Network-Based Fuzzy Logic Control and Decision System. *IEEE Trans. Computers* 40: 1320–1336.
8. Horikawa, S. I., Furuhashi, T. & Uchikawa, Y. (1992). On Fuzzy Modeling Using Fuzzy Neural Networks with the Back-Propagation Algorithm. *IEEE Trans. Neural Networks* 3: 801–806.
9. Jang, J. S. R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference Systems. *IEEE Trans. Syst., Man, Cybern* 23: 665–685.
10. Hauptmann, W. & Heesche, K. (1994). A Prototype of an Integrated Fuzzy-Neuro System. In *Proc. EUFIT'94 2nd European Congress on Fuzzy and Intelligent Technologies*, 1101–1104. Aachen, Germany, September 20–23.
11. Tien, B. T. & van Straten, G. (1995). Neural-Fuzzy Systems for Non-Linear System Identification – Orthogonal Least Squares Training Algorithms and Fuzzy Rule Reduction. In *Prepring of the 2nd IFAC/IFIP/EurAgEng Workshop on AI in Agriculture*, 249–254. Wageningen, The Netherlands, May 29–31.
12. Tien, B. T. & van Straten, G. (1996). Recursive Prediction Error Algorithm for the NUFZY System to Identify Nonlinear Systems. In *Proceedings of the 9th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE-96*, 569–574. Fukuoka, Japan, June 4–7.
13. Challa, H. (1989). Modeling for Crop Growth Control. *Acta Horticulturæ* 248: 209–216.
14. Chen, S., Cowan, C. F. N. & Grant, P. M. (1991). Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. *IEEE Trans. Neural Networks* 2: 302–309.
15. Udink ten Cate, A. J. (1985). Modeling and Simulation in Greenhouse Climate Control. *Acta Horticulturæ* 174: 461–467.
16. Tchamitchian, M., van Willigenburg, L. G. & van Straten, G. (1993). *Short Term Dynamic Optimal Control of the Greenhouse Climate*. Wageningen Agricultural University, Dept. of Agricultural Engineering and Physics, MRS-report 92-3, 22 pp.
17. Tap, R. F., van Willigenburg, L. G. & van Straten, G. (1995). Experimental Results of Receding Horizon Optimal Control of Greenhouse Climate. *Acta Horticulturæ* 406: 229–238.
18. Tap, R. F. (1996). Receding Horizon Optimal Control of Greenhouse Climate. In *Preparation*.
19. Wang, L. X. (1994). *Adaptive Fuzzy Systems and Control – Design and Stability Analysis*. Prentice-Hall: Englewood Cliffs, New Jersey.

Artificial Keys for Botanical Identification using a Multilayer Perceptron Neural Network (MLP)

JONATHAN Y. CLARK and KEVIN WARWICK

Department of Cybernetics, P.O. Box 225, University of Reading, Whiteknights, Reading, RG6 6AY, UK (E-mail: cybjyc@cyber.reading.ac.uk)

Abstract. In this paper, practical generation of identification keys for biological taxa using a multilayer perceptron neural network is described. Unlike conventional expert systems, this method does not require an expert for key generation, but is merely based on recordings of observed character states. Like a human taxonomist, its judgement is based on experience, and it is therefore capable of generalized identification of taxa. An initial study involving identification of three species of *Iris* with greater than 90% confidence is presented here. In addition, the horticulturally significant genus *Lithops* (Aizoaceae/Mesembryanthemaceae), popular with enthusiasts of succulent plants, is used as a more practical example, because of the difficulty of generation of a conventional key to species, and the existence of a relatively recent monograph. It is demonstrated that such an Artificial Neural Network Key (ANNKEY) can identify more than half (52.9%) of the species in this genus, after training with representative data, even though data for one character is completely missing.

Key words: Aizoaceae, expert system, identification, *Iris*, key, *Lithops*, Mesembryanthemaceae, multilayer perceptrons, neural network

1. Introduction

Traditional diagnostic keys for naming of taxa (biological entities) have long played a fundamental part with regard to practical biological identification. Bracketed or indented keys, dichotomous or otherwise have the advantage that they can appear on the printed page, but have a number of disadvantages. These include the high level of diagnostic skill and knowledge of specialised terminology needed, also sometimes considerable linguistic ability. In addition, when a couplet is particularly troublesome, it is often necessary to follow the trail both ways, adding to the confusion. One method to help alleviate these problems is by means of on-line computerized keys using an expert system approach, for example as shown by Pankhurst and Aitchison (1975), and Lobanov et al. (1981). In these cases, a human expert provided information on the relative diagnostic importance of the characters used, in order for the program to be developed. It is not necessary to go into further details here, as a good account is given by Pankhurst (1991). Artificial neural

networks have already been applied to the problem of identification of biological taxa with respect to pyrolysis mass spectrometry of microbial organisms (Goodacre 1994; Goodacre et al. 1992; Goodacre et al. 1994a, 1994b). The aim of this paper is to emphasise that neural networks are also useful for practical botanical (or zoological) keys, and to describe some advantages such keys may have over conventional methods for identification. Multilayer perceptrons (MLPs) (see Figure 1) are of particular interest here, as they have already been shown to be of value for identification of faults in industrial machinery (Clark and Warwick 1995; Matthews et al. 1995; Ray 1991) and medical diagnosis (Yoon et al. 1989). The ability of such methods to model non-linear multidimensional systems is also likely to be of significant value for identification of biological entities, whose definitions are somewhat fuzzy due to the natural variation within populations. Two case studies are presented here. The first uses only four continuous variables to distinguish between three species of *Iris*. The second study is a more ambitious practical effort to identify 34 species in the highly specialised succulent genus *Lithops*.

2. Materials and Methods

2.1 Case study: Identification of three species of *Iris*

For a pilot exercise, it was decided to use the Fisher-Anderson *Iris* data (Fisher 1936) that has already been extensively studied by Everitt (1993), in his work on cluster analysis. These data comprise measurements of four floral characters of fifty individuals of each of the taxa *Iris setosa* Pall. ex Link, *I. versicolor* Linn. and *I. virginica* Linn. The characters used are sepal length & width and petal length & width. Following the classification used by Mathew (1981), all three taxa belong to *Iris* subgenus *Limniris*, Section *Limniris*; *I. setosa* being in the Series *Tripetalae*, the remaining 2 taxa both belonging to Series *Laevigatae*. This set of data was chosen for this study primarily because a simple, but effective neural network key could be generated, using biological data that have already been extensively used for analysis. Whereas *I. setosa* is fairly readily distinguished from the other two, *I. versicolor* and *I. virginica* are more difficult to separate. Fisher's work is particularly interesting with relevance to that presented here, since he was able to maximise the difference between the three taxa by means of mathematical calculation, which is basically the aim of this study. Although Fisher was an accomplished statistician, and *I. setosa* was easily separated from the other two taxa, by means of linear discriminant functions, it was more difficult to distinguish *I. versicolor* from *I. virginica*, due to their more

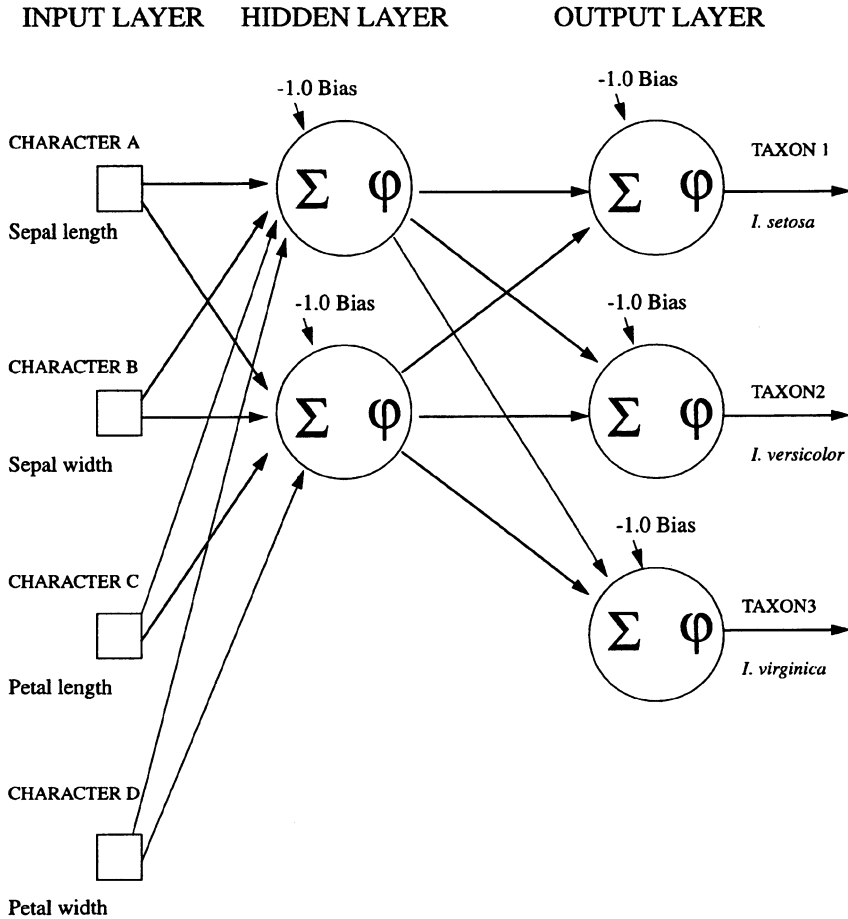


Figure 1. A multilayer perceptron IRIS ANNKEY.

similar floral characteristics. Indeed, he states that ‘... *there is some overlap of the distributions [of the evaluated results using the discriminant function], so that a certain diagnosis of these two species could not be based solely on these four measurements of a single flower taken on a plant growing wild*’ (Fisher 1936). Examination of the graphical results at the end of Fisher’s paper reveals that a maximum of 91.33% of the data set could theoretically be identified correctly by this method. Advanced techniques such as neural networks and expert systems were, of course, unknown at that time.

2.2 Case study: Identification of thirty-four species of *Lithops*

A more extensive study has also been carried out here on identification of species in the genus *Lithops*, which comprises about 34 species of succulent xerophytes native to South Africa and Namibia. This interesting genus of plants, popular amongst collectors of succulent plants, have a curious resemblance to stones, with attractive marbled and spotted leaf surfaces. There are a number of existing conventional keys to identification of this group, including those by De Boer and Boom (1964), Fearn (1981), and a new one by Clark (1996). However, the 'living stones' are extremely variable, it being difficult to choose consistent diagnostic characters on which to base the construction of a key. It was probably for this reason that Cole (1988), the latest monographer of the genus, declined to include a key. However, the excellent descriptions in this monumental work, representing a lifetime of dedicated study of plants in the field as well as in cultivation, are a good source of data on which to base the kind of analysis presented in this paper.

Table 1 shows the data extracted from Cole's monograph for use in training the network. Characters were chosen that were consistently described throughout the group, and that would be available to determine from living material during much of the year. An attempt was made to include large numbers of continuous variables, which can be measured relatively objectively by non-experts. The acronyms used in the table are identical with those used by Wallace (1990) in his study of allozyme variation in the genus, except here TER is used for *L. terricolor* (LOC used by Wallace was *L. localis*, which is the same taxon, and the correct earlier name), and HOO is used for *L. hookeri* (= TUR *L. turbiniformis*, used by Wallace). The species concepts used are those of Cole (1988), the data having been obtained from descriptions of the type variety or subspecies.

2.2.1 Description of characters selected

A *Lithops* plant consists of one or more pairs of leaves, each pair being fused for much of their length, the unfused portion forming a fissure or cleft dividing the leaves. These leaf pairs are called 'bodies' and may be grouped into clumps, caused by branching. The tops of the leaves are usually somewhat flattened, the transparent/translucent surface, or 'window' often being the only part of the plant showing above the surface in the wild. Each year, the fissure widens and a new pair of leaves emerge, the old pair shrivelling as the young leaves develop. The various colored branching lines and dots that are useful for identification are brightest and clearest on the new leaves, but these are often hidden for part of the year, making successful identification somewhat seasonal. The characters chosen here have been selected mostly because they

Table 1. Characters selected for training of Lithops ANNKEY

Species Name	Acronym	Body Profile	Body Top Curvature	Mean Fissure Depth (mm)	Reddish Dots	Reddish Lines	Dusky Dots	Face: minor axis (mm)	Face: major axis (mm)	Mean Body Clumps	Main Flower Color	Flower Centre Color	Flower Diameter (mm)	Sepal Number/Cell
<i>aucampiae</i>	AUC	truncate	conv/flat	5.0	absent	absent	difficult	25.0	35.0	3.5	Y	Y	35.0	6
<i>bromfieldii</i>	BRO	truncate	flat/conv	6.0	present	present	difficult	20.0	25.0	3.0	Y	Y	30.0	5
<i>comptonii</i>	COM	trun/cord	conv/flat	11.5	absent	absent	difficult	17.0	25.0	3.0	Y	W	22.5	5
<i>dinteri</i>	DIN	truncate	flat/conv	8.0	present	difficult	difficult	15.0	20.0	2.0	Y	Y	22.5	5
<i>divergens</i>	DIV	bicuneate	convex	9.5	absent	absent	difficult	12.0	20.0	3.0	Y	W	17.5	5
<i>dorotheae</i>	DOR	trun/cord	convex	8.5	present	present	difficult	13.0	20.0	3.5	Y	Y	27.5	5
<i>francisci</i>	FRA	cord/trun	conv/flat	9.0	absent	absent	present	17.0	24.0	4.5	Y	Y	17.5	5
<i>fulviceps</i>	FLV	truncate	flat/conv	6.5	present	present	present	23.0	30.0	2.5	Y	Y	27.5	5
<i>gesineae</i>	GES	cord/trun	convex	9.0	absent	absent	present	18.0	28.0	1.5	Y	Y	30.0	6
<i>geyeri</i>	GEY	cord/trun	convex	10.0	absent	absent	difficult	15.0	20.0	4.5	Y	W	27.5	5
<i>gracilidelineata</i>	GRA	truncate	conv/flat	5.5	present	present	difficult	25.0	35.0	1.0	Y	Y	30.0	6
<i>hallii</i>	HAL	truncate	flat/conv	5.5	present	present	absent	20.0	27.0	2.5	W	W	30.0	6
<i>helmutii</i>	HEL	cord/bicun	convex	12.5	absent	absent	absent	15.0	23.0	4.0	Y	W	27.5	5
<i>herrei</i>	HER	cord/trun	convex	11.0	absent	absent	difficult	17.0	23.0	4.5	Y	W	17.5	5
<i>hookeri</i>	HOO	truncate	flat/conv	5.0	difficult	difficult	difficult	23.0	30.0	3.0	Y	Y	30.0	6
<i>julii</i>	JUL	truncate	flat/conv	7.5	difficult	present	absent	20.0	25.0	3.0	W	W	30.0	5
<i>karasmontana</i>	KAR	truncate	flat/conv	7.0	difficult	present	difficult	20.0	25.0	4.0	W	W	30.0	5

Table 1. Continued

Species Name	Acronym	Body Profile	Body Top Curvature	Mean Fissure Depth (mm)	Reddish Dots	Lines	Dusky Dots	Face: minor axis (mm)	Face: major axis (mm)	Mean Body Clumps	Main Flower Color	Flower Centre Color	Flower Diameter (mm)	Sepal Number	Fruit Cell Number
<i>lesliei</i>	LES	truncate	flat/conv	3.5	absent	absent	absent	difficult	23.0	30.0	3.5	Y	Y	32.5	5
<i>marmorata</i>	MAR	trun/cord	convex	15.0	absent	absent	absent	difficult	17.0	25.0	4.0	W	W	30.0	6
<i>meyeri</i>	MEY	bicuneate	flat/conv	15.0	absent	absent	absent	difficult	18.0	27.0	2.5	Y	W	27.5	5
<i>naureniae</i>	NAU	bicun/cord	convex	14.0	absent	absent	absent	absent	20.0	25.0	3.5	Y	Y	27.5	5
<i>olivacea</i>	OLI	truncate	flat/conv	9.0	absent	absent	absent	absent	15.0	18.0	6.5	Y	W	30.0	5
<i>optica</i>	OPT	cord/trun	conv/flat	9.5	absent	absent	absent	absent	15.0	20.0	3.5	W	W	13.5	5
<i>otzeniana</i>	OTZ	cord/trun	convex	10.5	absent	absent	absent	absent	18.0	25.0	3.5	Y	W	25.0	5
<i>pseudotruncatella</i>	PSE	truncate	flat/conv	6.0	difficult	present	present	present	25.0	32.5	3.0	Y	Y	32.5	6
<i>ruschiorum</i>	RUS	cordate	convex	12.5	difficult	difficult	absent	absent	20.0	25.0	3.5	Y	Y	20.0	5
<i>salicola</i>	SAL	truncate	flat/conv	7.0	difficult	difficult	absent	absent	13.0	20.0	3.5	W	W	30.0	5
<i>schwantesii</i>	SCH	truncate	flat/conv	6.0	difficult	present	difficult	difficult	20.0	25.0	2.0	Y	Y	27.5	5
<i>terricolor</i>	TER	cordate	conv/flat	8.5	absent	absent	present	present	15.0	20.0	3.5	Y	Y/W	20.0	5
<i>vallis-mariae</i>	VAL	truncate	flat/conv	7.5	absent	absent	absent	absent	23.0	30.0	3.0	Y	Y	22.5	5
<i>verruculosa</i>	VER	truncate	flat/conv	6.0	present	difficult	absent	absent	18.0	25.0	3.0	Y/O	Y/O	22.5	5
<i>villetii</i>	VIL	trun/cord	flat/conv	10.0	absent	absent	difficult	difficult	18.0	23.0	3.0	W	W	22.5	6
<i>viridis</i>	VIR	cordate	convex	15.0	absent	absent	absent	absent	15.0	20.0	3.0	Y	W	27.5	5
<i>wernerii</i>	WER	cord/trun	convex	6.0	difficult	present	present	present	15.0	20.0	2.5	Y	Y	17.5	6

are relatively unambiguous and are measurable throughout much of the year. In this study, they have been treated as follows:

1. Body profile: shape of the leaf-pair body as viewed from the side e.g. cordate = heart-shaped
2. Body top curvature: this varies from flat to strongly convex
3. Fissure depth: measurement from the base of the fissure/cleft between the pair of leaves to the leaf apex
4. Reddish dots: also known as 'rubrications' on the top of the leaves (a $\times$ 10 hand lens may be necessary)
5. Reddish lines: linear rubrications on top of the leaves (again, a hand lens may be necessary)
6. Dusky dots: also known as 'pellucid' dots on the leaf top, small (< 1 mm) greyish, transparent, often indistinct dots, best visible in reflected light
7. Face minor axis: smallest dimension of leaf pair from above, measured at the fissure
8. Face major axis: largest dimension viewed from above. At times of year when leaf pair have separated to reveal new leaves, the top of the old leaves (if not significantly shrivelled) should be measured, less the gap caused by the new leaves.
9. Clumps: the number of individual plant bodies (leaf pairs) comprising one plant (not counting extra pairs on each body)
10. Main flower color: predominant color of the flower (Y: yellow; W: white; or rarely O: orange)
11. Flower centre color: color of the petals close to the flower centre (some flowers have a white centre)
12. Flower diameter: as measured when flowers are fully open
13. Sepal no./ Fr. cell no.: number of sepals in the calyx, which is the same as the number of cells (segments) in the fruit, and the number of stigma lobes.

Note that the non-continuous variables are treated as ordered multistate characters. Also, intermediate states are considered to be the same e.g. cordate/truncate = truncate/cordate. A character state is described as 'difficult' in the test data if it is very difficult to decide on the state e.g. whether or not 'dusky dots' are present. In the training data, it is used to indicate an intermediate generalized state, e.g. when the species is described as 'sometimes possessing dusky dots.'

2.3 Network parameters and methodology

The simple multilayer perceptron (MLP) used in this study was written at Reading University and uses one input node for each character, a variable number of hidden nodes (optimum currently found by experimentation), and one output node for each species to be identified. It is fully connected between layers, and contains one hidden layer. A representation of the architecture is presented in Figure 1. Although this diagram relates to the *Iris* study, the structure is similar for the *Lithops* study, differing only in the number of nodes in each layer. The input vectors were normalized between -1.0 and 1.0 , to reduce training times, these derived normalization vectors also being fed to the network for use during testing. The order of presentation of input vectors was randomised between epochs and a fixed learning rate of 0.25 and bias of -1.0 were used in each case study.

The non-linear activation function used was a sigmoidal logistic function φ given by

$$\varphi(v_j) = \frac{1}{1 + \exp(-v_j)}$$

where v is the weighted sum over n inputs for node j given by

$$v_j = \sum_{i=0}^n w_i \cdot x_i$$

where w is the weight and x is the input value.

The desired output values fed to the network during training were offset slightly (by 0.1) from the boolean values to accelerate the learning process (Haykin 1994). Therefore, an output of 0.9 from an output node meant 100% indication of that taxon, and 0.1 indicated 0% . During the test phase the outputs were normalized between 0 and 100 to give a percentage-like figure. The output node with the highest value was taken to be the 'winner', and the network was said to have identified that particular input vector as belonging to the species represented by that node.

The backpropagation algorithm used was the generalized delta rule, attributed to Rumelhart and McClelland (1986). The synaptic weights (w) were thus adjusted according to

$$w_{ji}(n+1) = w_{ji}(n) + \alpha[w_{ji}(n) - w_{ji}(n-1)] + \eta\delta_j(n)y_i(n)$$

where n is iteration number, η is the learning rate, α is the momentum and y is the node output $\varphi(v)$. For a node j in the output layer, δ is given by

$$\delta_j = y_j(1 - y_j)(t_j - y_j)$$

where t is the desired output for node j , y is the actual output for node j , For a node j in the hidden layer, δ is given by

$$\delta_j = y_j[1 - y_j] \cdot \sum_{k=1}^K [\delta_k w_{kj}]$$

where the sum is over all the K nodes in the output layer. Pattern mode training was used, based on the sum of squared errors (ϵ), given by

$$\epsilon = \frac{1}{2} \sum_{k=1}^K e_k^2$$

where e is the instantaneous error at output node k for that input pattern, and the sum is over all the K output nodes. The Mean Squared Error over the N records in the epoch is given by

$$MSE = \frac{1}{N} \sum_{n=1}^N \epsilon_n$$

Training was continued until a preset number of iterations (epochs) had been carried out. In the case of the *Lithops* study, a number of trials were carried out using different numbers of iterations.

2.3.1 *Iris* study

In the first exercise, using the *Iris* data, the original data set comprising 50 sets of measurements from each of the three species was first combined into a single file, the order of these records being shuffled randomly like a pack of cards. A total of 15 records (containing a mixture of all 3 classes) were then removed for testing purposes. The remaining 90% (135 records) were used to train the network. A fixed number of iterations (300) was used for this pilot study, the intention being to vary this if a good result was not achieved. The number of hidden nodes was varied in order to determine the best network architecture. A momentum value was not used, that is $\alpha = 0$ (see section 2.3).

2.3.2 *Lithops* study

The second exercise, was an attempt to recognise all 34 species in the genus *Lithops* as recognised by Cole (1988), and to provide a demonstration of how an ANNKEY could be produced merely from descriptions. A recently

discovered species, *L. coleorum* (Hammer and Uijs 1994), has been omitted from this study, as it is not yet common in collections, and was unknown when Cole wrote his monograph. Although *L. steineckiana* was included in the monograph, it is not considered in this study, as it is believed to be of hybrid origin. The training set consisted merely of 34 input vectors, each being used as an exemplar for one species. Each of these contained 13 character states extracted from the descriptions of the type variety of each species in the book, together with an indication of the species to which that record belonged. The principle was therefore to provide idealized examples on which to train the network. Ideally, measurements from each of the individual plants originally studied by Cole would form better input vectors. However, since these data were not available, a decision was made to carefully extract the character states of the hypothetical ideal representative of each species. This was considered to be adequate since Cole's descriptions are basically summaries of the characteristics of the species, based on a lifetime of study of these plants, both in the wild and in cultivation. Where integer numeric characters (e.g. number of fruit sections) were found to be variable, the modal value was used. The mean was substituted for continuously variable characters (e.g. flower diameter) where a range was given in the description. Not all characters described were selected; just those that were numeric or relatively unambiguous, consistently described, and easily coded into discrete states suitable for numeric processing.

It was decided that the best trial of the neural network method would be to test the trained network using real data obtained from live plants. The extreme succulence of these kinds of plants means that conventional herbarium specimens are often useless, as they are reduced to two dimensions, and most of the diagnostic colors and patterns quickly fade. Although the overall shape can be retained by preservation in spirit, most of the colors cannot. Thus keys to these plants are mostly written and used with regard to living specimens. These data, shown in Table 2, were obtained from plants of known species in the private reference collection owned by one of the authors (J. Y. Clark), at a time of year (late winter in the UK) when the states of the characters would be particularly difficult to determine, i.e. this was intended to be a severe test of the technique. Note that the states of a number of characters were unknown (designated '?'). In these cases, the mean of the values for that character used for training was substituted. For example, although data for the flower diameter were not available for the test phase, this was more realistic, since character states are often missing when attempting an identification. Character states in the test set that differ from that in the ideal training input vector are indicated by the use of italics. Where possible, the population (Cole Number) from which each test plant was descended has

been indicated. Further data on varietal status and location can be found in Cole's *Lithops Locality Data* booklet (Cole 1986).

The test set thus consisted of 34 records, each containing character states recorded from a living plant of known identity. This class information, although not provided to the network, was used to test the accuracy of the diagnosis achieved. For each set of network parameters, three trials (A, B and C) were carried out, and the proportion of the test set that were correctly identified was recorded. The number of nodes in the hidden layer, and the number of iterations were varied in order to determine the optimum configuration. During testing, the species corresponding to the output node with the highest output was taken to be the first choice of the network. The outputs of the second and third highest output values were also recorded to provide second and third choices, in order of preference, for the convenience of the user. The network output from these nodes effectively gives a measure of the relative confidence level of the decision made.

3. Results

3.1 *Iris study*

Table 3 shows the MSE and % successful identifications reached at the end of training and on presentation of the test (validation) data set. Although better results are achieved during training using 3 hidden nodes, the lowest MSE on validation is found using only 2 hidden nodes. Thus, the following observations relate to results obtained when only two nodes were used in the hidden layer (see Figure 1), as this configuration gave the best ability to generalize. After 300 iterations through the entire training data set, the mean squared error over the epoch had reduced to 0.0117, training having taken a mere 16 seconds on a Sun Sparc 5 UNIX workstation. At this stage, the network identified the training set to an accuracy of 97.04%. A curve showing the network learning is shown in Figure 2. Here, it can be seen that initial training was very rapid, later reducing to a much slower rate. The fast training is largely due to the small size of the data set. A larger data set would be expected to take a much longer time to train (up to 30 minutes or more, depending on the size of the data set). Testing of the trained network, which gives a measure of the ability of the network to identify previously unseen input vectors, gave 93.33% successful identifications over the epoch, with a confidence level greater than 90%. That is, for each successful identification, the normalized output of the winning output node was greater than 90. In other words, correct diagnosis can be achieved over 90% of the time using these neural network techniques. As extremely good results were achieved

Table 2. Data collected from living specimens for testing of Lithops ANNKEY

Species Name	Acronym /Cole Number	Body Profile	Body Top Curvature	Mean Fissure Depth (mm)	Reddish Dots	Reddish Lines	Dusky Dots	Face: minor axis (mm)	Face: major axis (mm)	Body Clumps	Main Flower Color	Flower Centre Color	Flower Diameter (mm)	Sepal/ Fruit Cell Number
<i>aucampiae</i>	AUC/C001	truncate	conv/flat	3.0	absent	absent	difficult	22.0	32.0	2	Y	Y	?	6
<i>bromfieldii</i>	BRO/C?	truncate	flat/conv	6.0	present	present	absent	18.0	26.0	5	Y	Y	?	5
<i>comptonii</i>	COM/C377	trun/cord	conv/flat	8.0	absent	absent	absent	14.0	21.0	2	Y	W	?	5
<i>dinteri</i>	DIN/C206	truncate	flat/conv	7.0	present	difficult	difficult	13.0	18.0	2	Y	Y	?	5
<i>divergens</i>	DIV/C201	bicuneate	conv/flat	13.0	absent	absent	absent	18.0	24.0	1	Y	W	?	?
<i>dorotheae</i>	DOR/C300	trun/cord	convex	9.0	present	present	absent	13.0	24.0	2	Y	Y	?	6
<i>francisci</i>	FRA/C140	cord/trun	conv/flat	10.0	absent	absent	present	14.0	20.0	4	Y	Y	?	?
<i>fulviceps</i>	FLV/C222	truncate	flat/conv	7.0	present	difficult	present	21.0	28.0	3	Y	Y	?	5
<i>gesineae</i>	GES/C078	truncate	flat/conv	11.0	absent	absent	present	24.0	33.0	1	Y	Y	?	7
<i>geyeri</i>	GEY/C274	cordate	convex	7.0	absent	absent	absent	14.0	21.0	2	Y	W	?	5
<i>gracilidelineata</i>	GRA/C309	truncate	conv/flat	6.0	difficult	difficult	absent	17.0	30.0	1	Y	Y	?	6
<i>hallii</i>	HAL/C039	truncate	flat/conv	6.0	present	present	absent	21.0	34.0	2	W	W	?	?
<i>helmutii</i>	HEL/C271	cord/bicun	convex	12.0	absent	absent	absent	17.0	24.0	3	Y	W	?	5
<i>herrei</i>	HER/C237?	cord/trun	conv/flat	11.0	absent	absent	absent	16.0	21.0	2	Y	W	?	5
<i>hookeri</i>	HOO/C118	truncate	flat/conv	5.0	difficult	difficult	absent	23.0	40.0	3	Y	Y	?	6
<i>julii</i>	JUL/C063	truncate	flat/conv	8.0	difficult	difficult	absent	18.0	25.0	5	W	W	?	6
<i>karasmontana</i>	KAR/C065	truncate	flat/conv	9.5	difficult	difficult	difficult	20.0	27.0	6	W	W	?	5

Table 2. Continued

Species Name	Acronym /Cole Number	Body Profile	Body Top Curvature	Mean Fissure Depth (mm)	Reddish Dots	Lines	Reddish Dots	Body	Face: minor axis (mm)	Face: major axis (mm)	Number in Flower	Main Color	Flower Centre Color	Flower Diameter (mm)	Sepal/ Fruit Cell Number
<i>lesliei</i>	LES/C010	truncate	flat/conv	4.0	absent	absent	absent	difficult	20.0	25.0	5	Y	Y	?	5
<i>marmorata</i>	MAR/C?	cordate	convex	13.0	absent	absent	absent	absent	17.5	25.0	2	W	W	?	6
<i>meyeri</i>	MEY/C212	cord/bic	flat/conv	14.0	absent	absent	absent	present	21.0	28.0	3	Y	W	?	?
<i>naureeniae</i>	NAU/C304	bicun/cord	convex	12.0	absent	absent	absent	absent	16.0	23.0	5	Y	Y	?	5
<i>olivacea</i>	OLI/C055	trun/cord	flat/conv	7.0	absent	absent	absent	absent	13.0	17.0	6	Y	W	?	5
<i>optica</i>	OPT/C081	cord/bic	conv/flat	10.0	absent	absent	absent	difficult	14.5	23.0	7	W	W	?	5
<i>otzeniana</i>	OTZ/C128	cordate	convex	8.5	absent	absent	absent	absent	17.0	24.0	6	Y	W	?	5
<i>pseudotruncatella</i>	PSE/C187	truncate	flat/conv	4.0	absent	difficult	present	23.0	33.0	2	Y	Y	Y	?	6
<i>ruschiorum</i>	RUS/C?	cordate	convex	14.0	present	present	difficult	25.0	30.0	1	Y	Y	Y	?	6
<i>salicola</i>	SAL/C086	truncate	flat/conv	6.5	absent	difficult	absent	absent	16.0	22.0	2	W	W	?	6
<i>schwantesii</i>	SCH/C143B	truncate	flat/conv	8.0	difficult	present	difficult	16.0	24.0	4	Y	Y	Y	?	6
<i>terricolor</i>	TER/C132	cordate	conv/flat	8.0	absent	absent	present	present	16.0	23.0	4	Y	?	?	5
<i>vallis-mariae</i>	VAL/C296	truncate	flat/conv	5.0	difficult	difficult	absent	absent	18.0	26.0	2	Y	Y	?	?
<i>veruculosa</i>	VER/C129	truncate	flat/conv	8.0	absent	difficult	absent	absent	17.0	29.0	2	Y/O	Y/O	?	5
<i>villetii</i>	VIL/C195	trun/cord	flat/conv	9.0	absent	absent	absent	absent	13.0	22.0	2	W	W	?	6
<i>viridis</i>	VIR/nearTL	cord/bic	convex	12.0	absent	absent	absent	absent	13.0	17.0	2	?	?	?	?
<i>wernerii</i>	WER/C188	cord/trun	convex	5.0	absent	absent	present	present	9.5	16.0	6	Y	Y	?	6

Table 3. Iris study – MSE and successes (% with first choice correct)

Hidden Nodes	After training		Validation	
	MSE	% Success	MSE	% Success
1	0.078	97.04	0.093	93.33
2	0.012	97.04	0.013	93.33
3	0.012	97.78	0.014	93.33

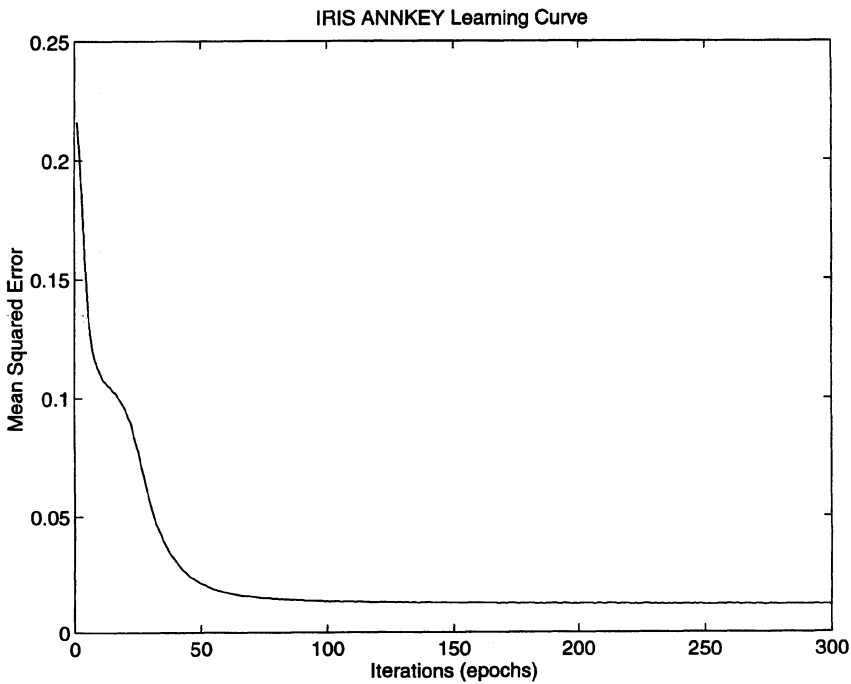


Figure 2. IRIS ANNKEY learning curve.

using 300 iterations, it was not considered necessary, in this pilot study, to vary the number of iterations.

3.2 Lithops study

After some trial and error, it was discovered that the generalization ability of the network was extremely sensitive to over-training (over-fitting), and a total of only 50 iterations with a learning rate of 0.25 and momentum 0.6 was found to be adequate, giving the highest mean success rate on validation. Training took less than 10 seconds on a Sun Sparc 5 workstation. The addition

of further iterations, although resulting in a reduction in mean squared error whilst training, caused a decrease in performance during the test phase. Table 4 shows the results of a number of validation trials using differing numbers of nodes in the hidden layer (H), and a range between 50 and 200 training iterations. Three trials (A, B and C) were carried out for each set of network parameters. Note that each trial involved separate training and the creation of a different trained network, so the mean values are important. After testing variation in the size of the hidden layer between the range of 16 and 56, the optimum number of hidden nodes was found to be 48, and the best number of iterations to be 50. That is, this gave best generalization when testing, and the highest mean % correct over the three trials with those network parameters. The following observations refer to this best result. The learning curve for the *Lithops* ANNKEY is given in Figure 3, the percentage of the taxa in the training set being learnt shown in Figure 4. After 50 iterations, all the training set (100%) were identified correctly. The results of presenting the test set to the *Lithops* network are presented in Table 5, with the target species emboldened. A winner-takes-all decision method was employed, with the three highest network outputs being taken to be the three favourite choices, in order of preference. Using this method a total of 52.94% of the test set were correctly identified as the first choice of the network. If the first and second preferences are considered, then 70.59% of the total are present in those 2 choices. This represents a considerable success since these species are well known to be difficult to identify, even using available conventional printed keys (DeBoer and Boom 1964; Fearn 1981; Clark 1996). Also, the test set represents a real test identification of living specimens at a time of year when many of the diagnostic colors and patterns are unclear. In addition, the measurements of one character, flower diameter, were not available at all since the flowers had long faded. The ANNKEY was, however, trained using data for that character, in order that it could be tested at a later date when data were available. Flower colors were still available from records made earlier in the year.

4. Discussion

The preliminary research involving the three species of *Iris* gave extremely encouraging results, suggesting that the MLP approach is useful as an alternative to statistical methods for use with continuous variable characters. Indeed better results (93.3%) are achieved using the neural network approach than the theoretical maximum (91.3%) achievable by the discriminant analysis method used by Fisher (1936). The *Lithops* study, by contrast, shows some of the problems that can occur with a more practical situation. Here, ordered

Table 4. Lithops study – Validation results (giving % with first choice correct)

Hidden Nodes	50 iterations				100 iterations				200 iterations			
	A	B	C	Mean	A	B	C	Mean	A	B	C	Mean
16	26.5	35.2	32.4	31.4	38.2	35.3	38.2	37.3	32.4	26.5	52.9	37.3
24	44.1	38.2	47.1	43.1	41.2	38.2	26.5	35.3	44.1	38.2	38.2	40.2
32	41.2	47.1	32.4	40.2	38.2	41.2	44.1	41.2	32.4	41.2	41.2	38.2
40	44.1	38.2	32.4	38.2	41.2	38.2	47.1	42.2	38.2	32.4	20.6	33.3
48	41.2	47.1	52.9	47.1	47.1	47.1	44.1	46.1	41.2	32.4	29.4	34.3
56	38.2	41.2	32.4	37.3	47.1	35.3	50.0	44.1	35.3	41.2	38.2	38.2

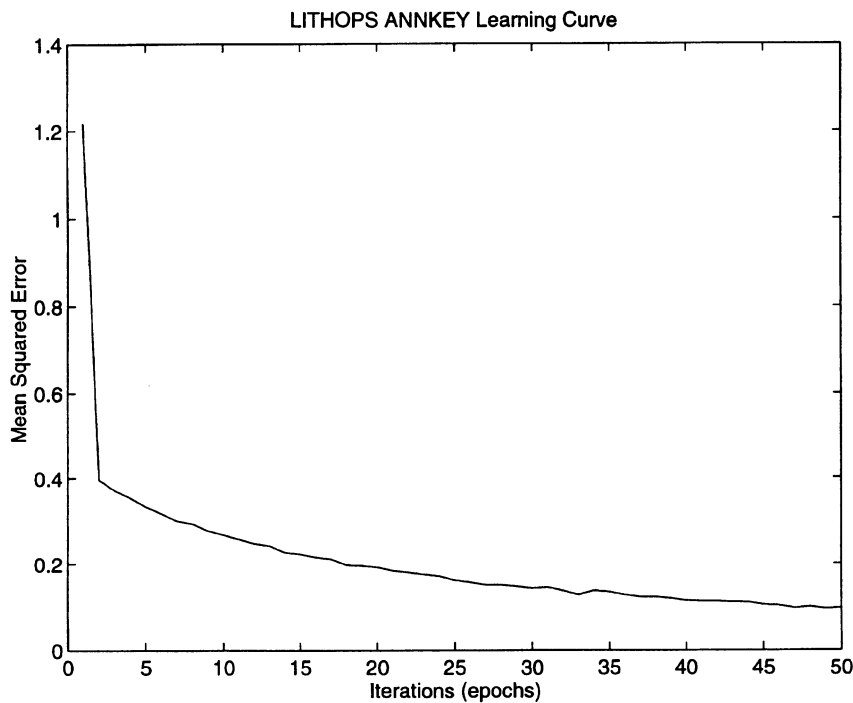


Figure 3. LITHOPS ANNKEY learning curve.

multistate characters and numeric characters have been used, some of which require very careful examination of the plants. Also, a correct identification is dependent on the validity of the existing classification. If any of the species are doubtfully distinct, then they will emerge as being difficult to identify by this or any other method, thus depressing the observed success rate. For instance, *L. naureniae* is closely related, and very similar to, *L. helmutii*. So it is not

Table 5. Lithops ANNKEY test analysis (showing three highest output values)

Species Name	Acronym/ Cole No.	Choice 1	Output	Choice 2	Output	Choice 3	Output
<i>aucampiae</i>	AUC/C011	AUC	53.66	HOO	15.06	VAL	13.27
<i>bromfieldii</i>	BRO/C?	VER	74.01	BRO	34.43	RUS	17.16
<i>comptonii</i>	COM/C377	COM	45.72	OLI	29.83	SCH	29.11
<i>dinteri</i>	DIN/C206	DIN	88.79	TER	18.20	VER	12.46
<i>divergens</i>	DIV/C201	MEY	61.70	NAU	25.61	SCH	14.50
<i>dorotheae</i>	DOR/C300	GRA	50.01	WER	35.30	DOR	29.98
<i>francisci</i>	FRA/C140	FRA	50.75	TER	36.15	OLI	16.92
<i>fulviceps</i>	FLV/C222	FLV	72.15	DIN	39.10	TER	29.97
<i>gestineae</i>	GES/C078	AUC	56.73	GES	34.25	PSE	22.14
<i>geyeri</i>	GEY/C274	DIV	26.39	OTZ	22.95	HEL	16.94
<i>gracilidelineata</i>	GRA/C309	GRA	48.55	VER	37.30	SCH	21.41
<i>hallii</i>	HAL/C039	HAL	70.98	VER	23.40	GRA	21.28
<i>helmutii</i>	HEL/C271	NAU	31.46	HEL	21.54	DIV	12.80
<i>herrei</i>	HER/C237?	COM	44.05	OLI	24.09	SCH	21.44
<i>hookeri</i>	HOO/C118	VAL	63.58	GRA	49.12	VER	42.34
<i>julii</i>	JUL/C063	HAL	57.18	VIL	27.17	OLI	26.67
<i>karasmontana</i>	KAR/C065	KAR	65.13	FLV	26.26	HER	23.88

Table 5. Continued

Species Name	Acronym/ Cole No.	Choice 1	Output	Choice 2	Output	Choice 3	Output
<i>lesliei</i>	LES/C010	VAL	30.52	LES	27.92	TER	20.48
<i>marmorata</i>	MAR/C?	MAR	24.33	RUS	22.30	OPT	15.02
<i>meyeri</i>	MEY/C212	MEY	51.64	TER	23.52	COM	21.70
<i>naureniae</i>	NAU/C304	HEL	25.94	NAU	22.69	OTZ	12.47
<i>olivacea</i>	OLI/C055	OLI	85.56	TER	23.36	VER	19.81
<i>optica</i>	OPT/C081	KAR	51.20	OPT	46.96	MEY	19.54
<i>otzeniana</i>	OTZ/C128	OTZ	31.64	HER	26.25	VER	25.53
<i>pseudotruncatella</i>	PSE/C187	PSE	35.52	AUC	34.97	GES	10.80
<i>ruschiorum</i>	RUS/C?	RUS	84.90	GRA	78.43	WER	31.97
<i>salicola</i>	SAL/C086	VIL	67.49	HAL	22.35	TER	13.20
<i>schwantesii</i>	SCH/C143B	WER	59.79	OLI	29.17	HOO	15.77
<i>terricolor</i>	TER/C132	TER	81.50	FRA	18.07	KAR	14.09
<i>vallis-mariae</i>	VAL/C296	VER	50.18	SCH	26.46	VAL	22.47
<i>verruculosa</i>	VER/C129	VAL	48.48	SCH	42.91	VER	12.20
<i>villetii</i>	VIL/C195	VIL	73.56	OPT	21.19	SAL	19.39
<i>viridis</i>	VIR/near Typ.Loc.	VIR	55.06	OPT	34.80	RUS	20.13
<i>wernerii</i>	WER/C188	WER	47.78	OLI	45.38	GEY	23.12

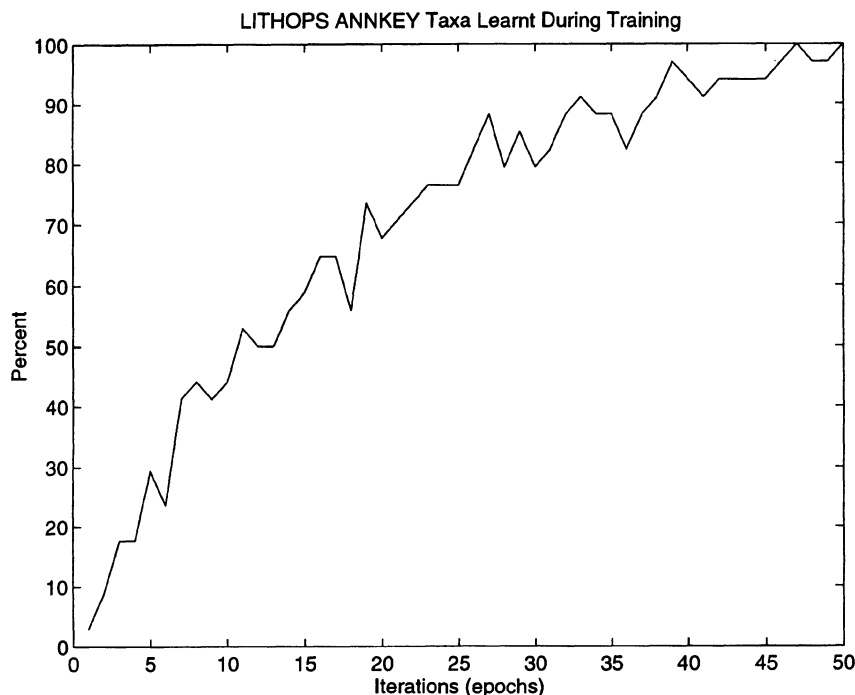


Figure 4. LITHOPS ANNKEY Taxa learnt during training.

surprising that the network 'misidentifies' NAU as HEL (for first choice), and vice versa, although the second choices are correct. Thus, as is the case with the construction and use of any other type of key, sometimes suggestions are made regarding possible inadequacies in the existing classification scheme. On the other hand, the ANNKEY makes some very surprising judgements, for instance misidentifying OPT as KAR (although the second choice is given as OPT). These species are quite unlike each other, except that they both have white flowers. It is possible that the addition of further characters would help with this kind of problem. However, some of the problems that the network has had are quite reasonable. For instance, *L. hallii*, *L. julii*, *L. salicola*, and to a lesser extent *L. villetii*, form a closely related group of possibly indistinct species; thus SAL is understandably misidentified as VIL or HAL, and JUL as HAL or VIL.

5. Conclusions and Further Work

This study has demonstrated that artificial neural networks are a useful and innovative alternative method for creating computer-based self-learning keys (ANNKEYs). Although further research is still necessary, these are of potential use in herbaria and museums, or even in the field, using portable computers. ANNKEYs have an advantage in that no 'expert' is needed, except for initial determination of characters. They learn in the same way as a human taxonomist, that is by experience. As further data becomes available, an ANNKEY can be updated by teaching it the new data, with a revision of the old data (just like the method used by a human taxonomist). ANNKEYs are of particular use when quantitative characters are used because they can be scored relatively easily and unambiguously. However, binary or ordered multistate characters may be used, if coded in a similar way to that used for cluster analysis. In the case of 'difficult' genera, where the delimitation of sub-taxa is less clear, such as in the genus *Lithops*, ANNKEYs can be used to generate not only 'favourites', but also second or third choices, together with a relative confidence rating of the identification. Although missing data can be a serious problem with conventional keys, this is only severe when using an ANNKEY if the missing characters are critical for separation of taxa. Although further work is necessary, unpublished preliminary results using the DELTA expert system (Dallwitz 1974, 1980; Dallwitz et al. 1993; Partridge et al. 1993) to produce a key, using the *Lithops* data from this study, indicate a success rate of 39.9%, whereas the ANNKEY has an accuracy of 52.94%. In the future, it is planned to continue the *Lithops* study using more characters, and using a more comprehensive test set in order to further test the generalization of the network, for example with flower diameter data available. Further studies using additional characters have already recently resulted in the construction of a new conventional dichotomous key to the genus (Clark 1996).

References

- Clark, J. Y. (1996). A key to *Lithops* N.E.Br. (Aizoaceae). *Bradleya* **14**: 1–9.
- Clark, J. Y. & Warwick, K. (1995). Detection of faults in a high speed packaging machine using a multilayer perceptron (MLP). *IEE Colloquium: Innovations in manufacturing control through mechatronics*, Newport, Gwent, UK, Digest No. 95/214: 7/1–7/3.
- Cole, D. T. (1986). *Lithops Locality Data*. Desmond T. Cole: Swakaroo, Emmarentia, South Africa, January.
- Cole, D. T. (1988). *Lithops, Flowering Stones*. Randburg, South Africa: Acorn Books.
- Dallwitz, M. J. (1974). A flexible computer program for generating identification keys. *Systematic Zoology* **23**: 50–57.
- Dallwitz, M. J. (1980). A general system for coding taxonomic descriptions. *Taxon* **29**: 41–46.

- Dallwitz, M. J., Paine, T. A. & Zurcher, E. J. (1993). *User's guide to the DELTA system: a general system for processing taxonomic descriptions*, 4th edition. Canberra, Australia: CSIRO Division of Entomology.
- DeBoer, H. W. & Boom, B. K. (1964). An analytical key for the genus *Lithops*. *National Cactus & Succulent Society Journal* **19**: 34–37, 51–55.
- Everitt, B. S. (1993). *Cluster Analysis*. New York: Edward Arnold/Halsted Press.
- Fearn, B. (1981). *Lithops*. Oxford, UK: British Cactus & Succulent Society (Handbook No. 4).
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics* **7**: 179–188.
- Goodacre, R. (1994). Characterisation and quantification of microbial systems using pyrolysis mass spectrometry: introducing neural networks to analytical pyrolysis. *Microbiology Europe* **2**(2): 16–22.
- Goodacre, R., Kell, D. B. & Bianchi, G. (1992). Neural networks and olive oil. *Nature* **359**: 594.
- Goodacre, R., Trew, S., Wrigley-Jones, C., Neal, M. J., Maddock, J., Ottley, T. W., Porter, N. & Kell, D. B. (1994a). Rapid screening for metabolite overproduction in fermentor broths, using pyrolysis mass spectrometry with multivariate calibration and artificial neural networks. *Biotechnology and Bioengineering* **44**: 1205–1216.
- Goodacre, R., Neal, M. J., Kell, D. B., Greenham, L. W., Nobel W. C. & Harvey, R. G. D. (1994b). Rapid identification using pyrolysis mass spectrometry and artificial neural networks of *Propionibacterium acnes* isolated from dogs. *Journal of Applied Bacteriology* **76**: 124–134.
- Hammer, S. A. & Uijis, R. (1994). *Lithops coleorum* S.A. Hammer & R. Uijis sp. nov., a new species of *Lithops* N.E.Br. from the Northern Transvaal. *Aloe* **31**(2): 36–38.
- Haykin, S. (1994). *Neural networks – a comprehensive foundation*. New York: Macmillan College Publishing Company, Inc.
- Lobanov, A. J., Schilow, W. F. & Nikritin, L. M. (1981). Zur Anwendung von Computern für die Determination in der Entomologie. *Deutsche Entomologie Zeitung* **28**: 29–43.
- Mathew, B. (1981). *The Iris*. London: B.T. Batsford Ltd.
- Matthews, C. P., Clark, J. Y., Sharkey, P. M. & Warwick, K. (1995). A comparison of cluster analysis and neural networks for the reliability of machinery. *Proceedings SPIE Conference Photons East*. Philadelphia.
- Pankhurst, R. J. (1991). *Practical Taxonomic Computing*. UK: University of Cambridge Press.
- Pankhurst, R. J. & Aitchison, R. R. (1975). A computer program to construct polyclaves. In Pankhurst, R. J. (ed.) *Biological Identification with Computers*, 73–78. London and Orlando: Academic Press.
- Partridge, T. R., Dallwitz, M. J. & Watson, L. (1993). *A primer for the DELTA system*, 3rd edition. Canberra, Australia: CSIRO Division of Entomology.
- Ray, A. K. (1991). Equipment fault diagnosis – A neural network approach. *Computers in Industry* **16**: 169–177.
- Rumelhart, D. E. & McClelland, J. L. (1986). *Parallel Distributed Processing*, Vols. 1 & 2. Cambridge, Mass.: MIT Press.
- Wallace, R. S. (1990). Systematic significance of allozyme variation in the genus *Lithops* (Mesembryanthemaceae). *Mitt. Inst. Allg. Bot. Hamburg: Proceedings of the twelfth plenary meeting of aetfat. Symposium VI*, 509–524. Hamburg, Germany: Band 23b.
- Yoon, Y., Brobst, R. W., Bergstresser, P. R. & Peterson, L. (1989). A desktop neural network for dermatology diagnosis. *Journal of Neural Network Computing* **1**: 43–52.

Video Grading of Oranges in Real-Time

MICHAEL RECCE¹, ALESSIO PLEBE², JOHN TAYLOR³ and
GIUSEPPE TROPIANO²

¹*Department of Computer and Information Science, New Jersey Institute of Technology,
University Heights, Newark, New Jersey, USA (email:recce@cis.njit.edu)*

²*Agricultural Industrial Development (AID), Industrial Park, Blocco Palma I, Catania, Sicily
(email: aid@cres.it)*

³*Department of Anatomy and Developmental Biology, University College London, Gower
Street, London WC1E 6BT, U.K.*

Abstract. We describe a novel system for grading oranges into three quality bands, according to their surface characteristics. The system is designed to process fruit with a wide range of size (55–100 mm), shape (spherical to highly eccentric), surface coloration and defect markings. This application requires both high throughput (5–10 oranges per second) and complex pattern recognition. The grading is achieved by simultaneously imaging each item of fruit from six orthogonal directions as it is propelled through an inspection chamber. In order to achieve the required throughput, the system contains state-of-the-art processing hardware, a novel mechanical design, and three separate algorithmic components. One of the key improvements in this system is a method for recognising the point of stem attachment (the calyx) so that it can be distinguished from defects. A neural network classifier on rotation invariant transformations (Zernike moments) is used to recognise the radial colour variation that is shown to be a reliable signature of the stem region. The succession of oranges processed by the machine constitute a pipeline, so time saved in the processing of defect free oranges is used to provide additional time for other oranges. Initial results are presented from a performance analysis of this system.

Key words: pattern recognition, multi-processor, image processing

1. Introduction

Most of the processing of fresh fruit in packing houses is highly automated. Machines are used effectively for operations such as washing, waxing, sorting by size and color, and packing. However the most important step in the process, namely inspection and grading in quality, is still, with few exceptions, performed manually throughout the world.

This step has not yet been fully automated in production systems, as it requires fast and complex image analysis. Although grading of fruit shares several common features with more classical automated inspection of manufactured goods, this problem is significantly more difficult due to the wider range in variation found in natural products. Automation of the grading process is expected to reduce the cost of this important step and to lead

towards the standardization of grades of fruit that is desired by international markets. The inspection process must grade individual oranges into a small number of quality bands (typically three). Oranges are assessed according to surface characteristics including: discoloration, bruising and other blemishes. The grade is a measure of the number and size of these surface marks. The process is further complicated by the fact that, in an image, the stem is difficult to distinguish from defects. Part of the difficulty in correctly grading oranges is the process of detecting and distinguishing the stem from other potential marks that determine the appropriate grade.

Several manufacturers have developed machines for grading oranges, but none of these incorporates a stem detection mechanism, and in general the bin selection process is less than precise. So far none of the existing machines is able to match the requirements set by the market. We know of only one prior description of an orange grading machine in the literature (Maeda 1987), but similar image processing techniques have been used to grade a wide range of fruit and vegetables (see Tillett 1991 for a review).

This paper presents a new approach, largely based on the use of neural networks to achieve a more thorough analysis of the surface of oranges, including the detection of the stem. This analysis is aided by a new mechanical design which provides simultaneous images of the fruit from six orthogonal directions. More traditionally the grading decision is based on one or more views from a single direction as the inspected object is moved under a fixed camera (Davenel et al. 1988; Shearer and Payne 1990). One of the key aims addressed by this video grading machine is the real-time performance required by the application. The detailed video-based analysis is computationally demanding, and the problem is compounded by the high throughput required for a commercial grading machine.

The key to the solution is the use of a flexible process, in which time consuming computations are only invoked when necessary. In the first step of the process a fast global analysis is initially applied to every view of each orange. Further, more detailed and time-consuming analyses are issued only when required as a result of the previous stages. As a consequence the processing time for an orange is highly variable. In general, the process is faster for high quality oranges, and the time required scales roughly with fruit size. To take advantage of this time variation, the grading decision and activation of the separating device is delayed so that processing time can be best apportioned between the stored images of fruit currently in the pipeline. Since the largest percentage of fruit is of good quality, the time frame between adjacent oranges can be much smaller than the actual time spent to perform the most detailed analysis for a worst-case orange. The effect of this is to improve the throughput and overall grading performance (proportion of

oranges correctly graded). However, correct grading is not guaranteed in a case in which many consecutive oranges all exhibit difficult surface features.

Even with careful time allocation the target performance can only be achieved if the processors are sufficiently fast. The video grading machine incorporates a commercially available digital signal processor (DSP) and a new specialized neural network parallel processor currently under development at Philips Electronics Laboratories (Maudit et al. 1992).

All of the layers of analysis used by the grading algorithm are based on neural networks, fed by different feature extractors. This application is particularly well suited for neural network based techniques, because it is difficult to define any geometrical or spectral properties for the natural surface characteristics of fruit (stem, discoloration, bruising and other blemishes). However, the human eye is able to distinguish between a good orange, a defect, or a stem quite easily, which shows that the visual appearance, at modest resolution, contains the necessary information for classification.

2. System Overview

The automated fruit inspection machine described here consists of three principal parts: a system to convey the fruit into the vision chamber, the vision chamber itself, along with the image processing system, and finally the mechanism which diverts the oranges into different lanes according to their estimated quality (Figure 1). Although the most important part of the system is the image processing system, the method used to obtain the images is not at all trivial, and its development has been closely linked to the processing algorithms. The basic problem is to inspect, at high speed, the complete surface of a nearly spherical object. The wide range of sizes and shapes found in oranges preclude the use of conventional conveyors (rollers and belts). A novel design has been developed, permitting the simultaneous inspection of the entire surface of an orange during a short flight through the vision chamber.

2.1. *Impeller*

The first stage of this new system is a mechanism for accelerating the oranges to a suitable velocity, spacing them apart from each other, and propelling them along a fixed trajectory through the inspection chamber. The idea here is that each orange should follow roughly the same path independent of its geometrical properties. In the present design this occurs as long as the contribution of air resistance is minimal, the orange is not rotating when it

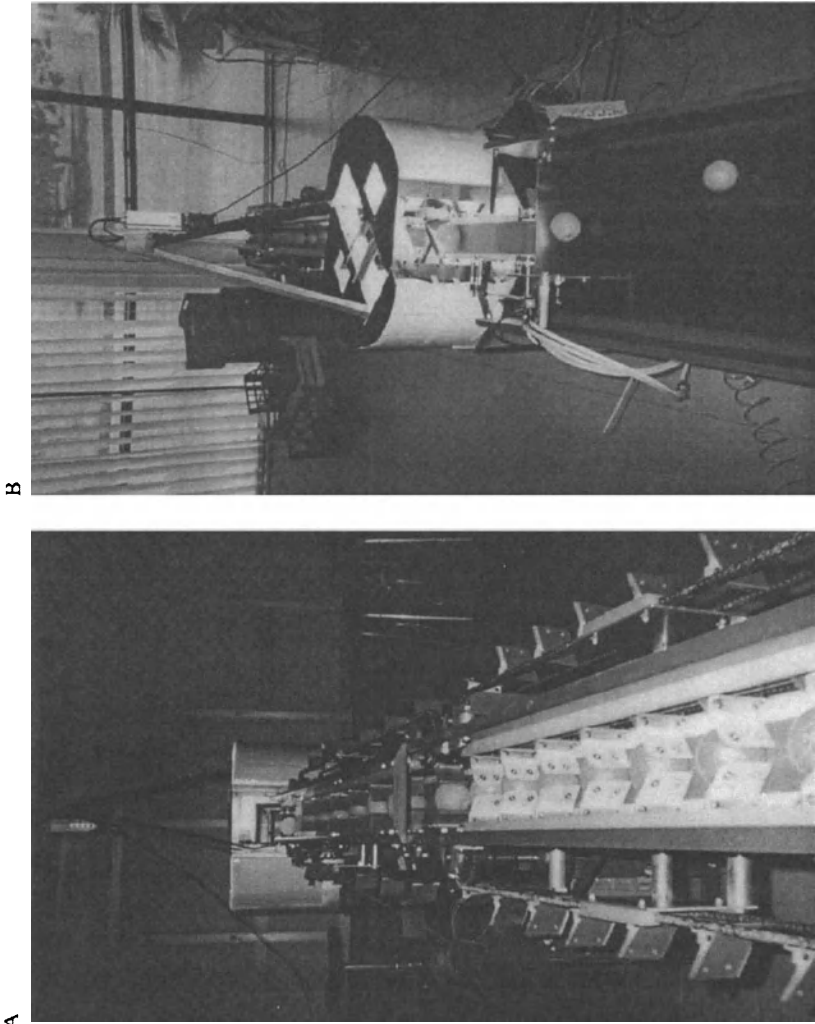


Figure 1. The prototype video grading machine. Part A shows oranges being carried by the impeller into the vision chamber. Note that there are two stages of acceleration, and in each of these the spacing between the oranges increases. Part B is a view of the oranges as the exit from the vision chamber and are sorted into three grades.

leaves the impeller and there is good control of all of the components of its initial velocity.

Also only one orange can be in the vision chamber at any point in time (otherwise it is impossible to obtain views of the front and the back of the orange). At prior stages in the packing house the oranges flow with a high packing density (approximately 100 mm center-to-center separation) at roughly ten oranges per second in an individual line. With a sequence of acceleration steps the speed of the oranges is increased by a factor of four and the spacing is increased by the same amount. The orange leaves the impeller with a speed of approximately 4 meters per second, travels 250 mm through the air, and lands on a belt that is moving with the same speed as the horizontal component of the initial velocity. The small change in velocity experienced by the orange minimizes the chances of bruising. The six views of each orange are captured simultaneously at the center of the vision chamber (see Figure 2).

2.2. Image capture and processing

The system hardware is VME-bus based, with four basic components: (1) the master processor (MC68030), (2) the color frame grabber based on a Texas Instruments TMS320C40 DSP, (3) the Philips prototype board, based on the L-Neuro2 parallel neural network processor and (4) an industrial digital interface board for synchronization and actuator commands.

The amount of processing required to make a decision on the quality of an orange depends on the quantity, type and distribution of defects. As described below a perfect orange can be processed very quickly, while a complex arrangement of defects requires a considerably longer time to process. For this reason we have constructed a multiple stage processing sequence, in which decisions that require the least time are performed first, effectively reducing the processing required at later stages. Furthermore, as described below the grading decision for an orange is delayed so that time saved in processing one orange can be applied to provide more time to process a subsequent orange.

There are three identified stages in the grading decision (identified below as separate processes). A fourth process provides the global supervision and control of the other three processes. The controlling process executes on the master processor (MC68030), supervises the other three processes, and keeps track of the location of individual oranges that are currently within the machine. The image processing steps are executed by the DSP and the L-Neuro2 (see (Maudit et al. 1992 for a description of the L-Neuro2).

In order to image the entire surface of the orange, one strategy is to use six planar views normal to the axes of a Cartesian coordinate system located at the

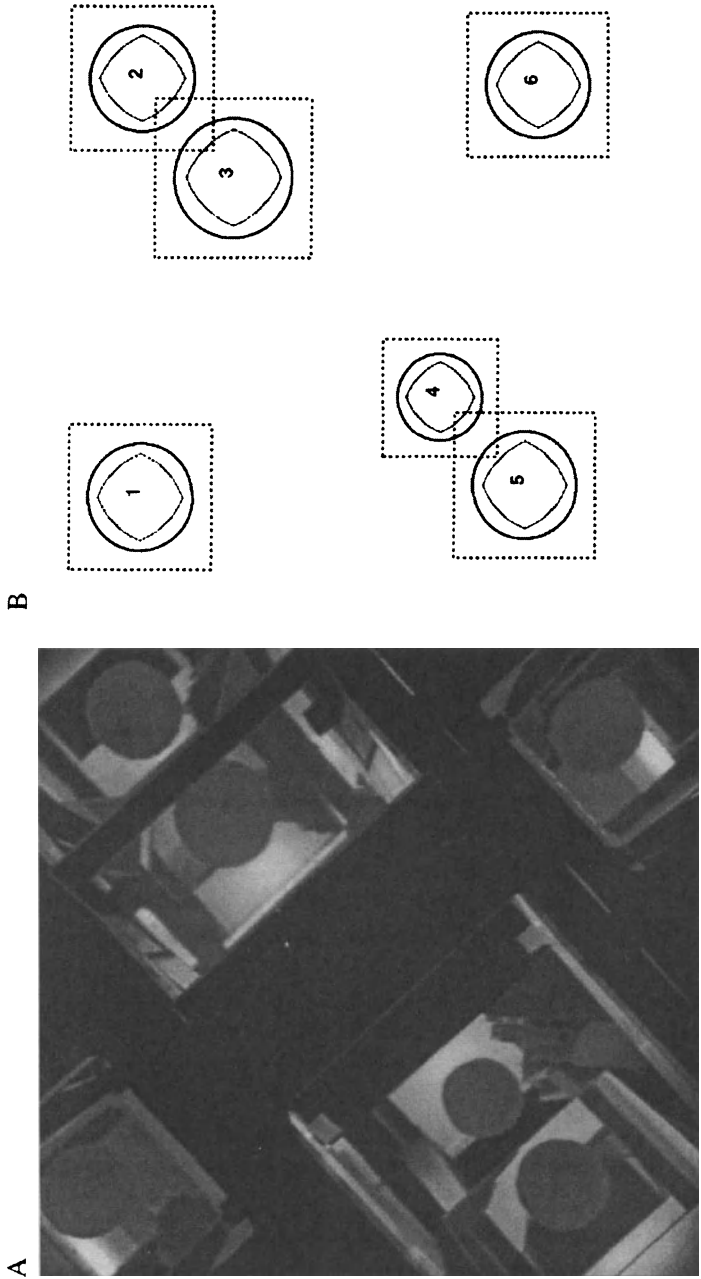


Figure 2. Six simultaneous views of an orange collected in the inspection chamber. The views are constructed using a set of mirrors. Part A shows an example image captured by the camera, and part B shows the regions extracted from the image. Color is used to segment the oranges from the background (Grasso and Recce 1996). In each view only the unique part is extracted as shown by the rounded shape that is drawn inside each of the circles.

center of the roughly spherical orange. The entire surface could be viewed, in theory, using two projections, but this results in considerable distortion from the curvature of the orange. If we assume that the orange is spherical than with six views the angle between the surface normal vector and the image plane is less than 45 degrees for over 95% of the surface of the orange. Six views are also quite practical to implement.

In the current prototype machine the six views are captured by projecting the image from a set of mirrors to an asynchronously triggered CCD camera. The camera is positioned directly above the center of the vision chamber, and the image capture is triggered when the orange is in mid flight in the center of the chamber. From the camera view point, four of the mirrors are in a "X" pattern and project four side views upwards. Two mirrors are used to project the underside of the orange upwards, and two mirrors are used to project the top view of the orange into the camera image. Figure 2 shows the camera view of the projected images. The image of the orange is the same size in each of the four side views, is slightly smaller (17%) in the underside view and is slightly larger (13%) in the top view. The figure also shows the region extracted from the six simultaneous views. The image chamber is uniformly illuminated with high speed fluorescent lights, but the chamber is designed so that light bulbs are not visible in any of the projected images.

In prior machines, a single image plane is used and the oranges are rolled and moved under the camera. In this case, the amount of surface seen, and the number of times that a surface element was seen, depended on the radius and eccentricity of the orange.

As soon as the image has been captured, the unique pixels are extracted from each of the six views of the orange. The symmetry of the viewing arrangement permits a calculation of the most appropriate regions from the six (assumed circular) views of the orange. The shape could be constructed by projecting the edges of a cube on to a circumscribed spherical screen, starting at a point at the center of the sphere, and then projecting this three dimensional curved cube onto a plane that is parallel to a face of the original cube. In addition, for each point selected from the image, a weighting factor is calculated which may be used to compensate for the viewing angle. Therefore, since points towards the edge of the orange are viewed more obliquely, a weighting value greater than 1.0 (ranging to approximately 1.6) is used to give each part of the orange surface equal value in calculations of the surface intensity histogram. The initial image extraction is performed by the TMS320C40 DSP.

Afterwards, each of the views is examined, and a view is excluded from further processing if it is unlikely that the particular view contains either defects or the stem of the orange. This decision is performed by a neural network algorithm that classifies color histograms (normalized red and green)

of the pixels contained in the view (described in more detail below). The DSP then passes each of the remaining views to the L-Neuro board for a more detailed analysis of local areas of the surface.

The image is scanned using a region based operator and each region is classified as defect free or not, using a second neural network (described below). The fraction of the surface area of the orange containing defects is the essential basis for the grading decision. Before treating an identified region as a defect, it is necessary to insure that it is not a stem. This is only done if the identified region is large enough to potentially be the stem. This is the most complex process in the grading operation and is carried out using a third neural network.

2.2.1. *Sorting system*

After the images of the orange have been obtained and the grading processing completed, the orange is deflected to the appropriate bin using purpose built pneumatic valves that release shaped jets of compressed air across the direction of motion and deflect the fruit into three separate areas.

During the visual analysis of each fruit, its dimensions, and hence mass are measured, which is used to set the timing and shape of the air jets. This activation is delayed to allow more fruit to be in a pipeline of process at the same time, and thus take advantage of the time-varying processing for each one.

3. Histogram Analysis

The first stage in the image processing is the classification of the color distribution of each of the views of an orange. Histograms of the normalized pixel values are constructed for two of the three color planes (red and green). Prior to this step, the color vectors are normalized to remove the effect of slight variations in the lighting. From empirical evidence, views of oranges that do not contain a stem or defects are found to have normally distributed red and green color components. This is to be expected, since the natural skin pigmentation is composed mostly of one color. Although, the peak color and its spread vary over different varieties of oranges, as well as over the season for the same variety.

Stems, discoloration, bruising and other blemishes tend to corrupt the smooth normal distribution of both the red and green pixel values. Unfortunately, this effect is often quite small, and the number of pixels with abnormal colors is unpredictable. Furthermore, in some cases a pigmentation gradient is present in good oranges, which may be caused by non-uniform exposure to sunlight during the ripening process.

Therefore, in part the classification of a view can be computed by the fit of the frequency distribution of the pixel values in the red and green color planes to a Gaussian function. Data which fit well to a Gaussian function are not likely to contain defects or the stem. The error is the summed difference between the best fit function and the histogram data. Figure 3 shows example best fit histograms for two defect free views, one view that contains a stem and one view that contains a defect. The examples in this figure show that while there is usually a good fit in the defect free images, there is a degree of variation. Also the view containing a stem and the view containing a defect can not be distinguished using the histogram analysis.

The surface defects on the oranges produce characteristic ranges of error in specific segments of the histogram. In order to evaluate the differences, the histogram data is divided into a set of segments, and in each segment I_i is computed to be the sum of the absolute value of the difference between the histogram data and the Gaussian fit to the data.

Empirically we found that a simple scoring rule based on the fit of a Gaussian function to the original histogram does not perform as well as a neural network based classifier that is used to learn the characteristic differences in a measured distribution from a normal distribution. The neural network algorithm used here is a modified form of back-propagation training of a multi-layer perceptron (Rumelhart et al. 1986; Werbos 1974) with ten inputs and two outputs. The input layer combines information from the red and green pixel color component histograms. There are two output classes: (1) no defects, or (2) either defects or a stem is present. Views of oranges that fall into the second class are passed to the second stage of the process, in which masks are used to analyze the surface in more detail. The number of input neurons, the partitioning of the histograms, and the number of hidden units have been empirically evaluated and compared.

As with other classification procedures, this procedure incorrectly classifies a fraction of the top quality oranges (Quality I) into a lower quality band (Quality II and Quality III), and places a fraction of the lower quality oranges into the top quality band. From the commercial point of view the full system should make fewer errors in downgrading the quality than in upgrading the quality. However, in this first stage of the processing it is safer to bias the process towards downgrading, and to run the additional processing stages when there is the possibility that a view of the orange contains a defect.

As suggested by Hush and Horne (1993), better generalization is achieved if the number of training samples is at least ten times larger than the number of weights in a multi-layer back propagation network. This neural network, and the neural networks at later stages of processing, is trained from a database that contains over 2000 examples (including a 20% test set).

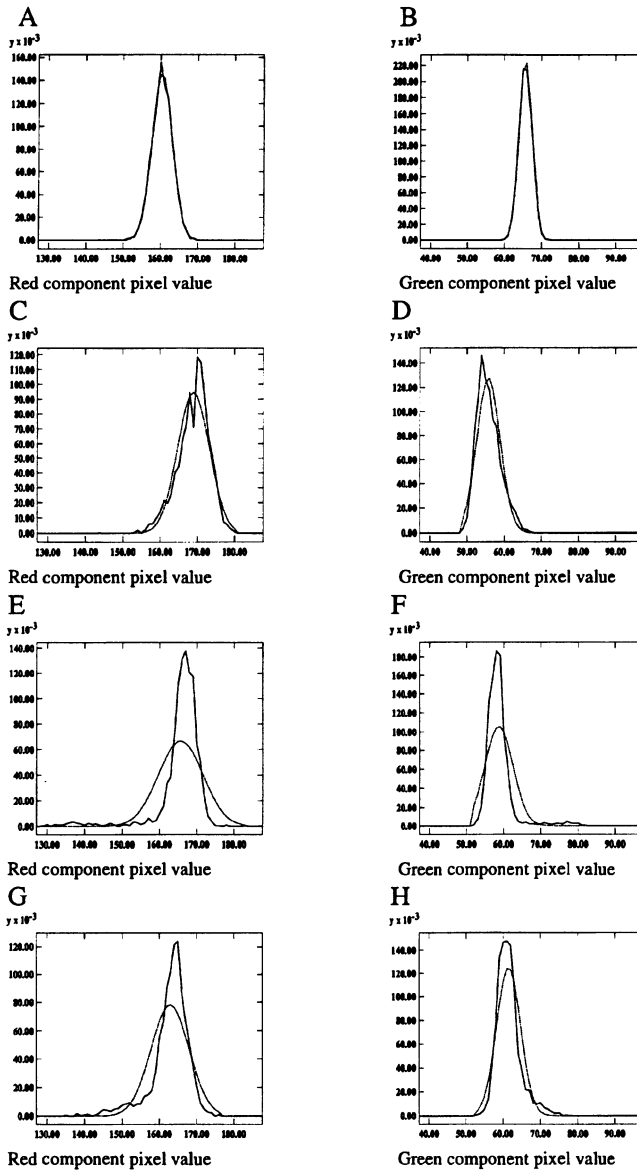


Figure 3. The red and green color distributions of four example views of oranges (solid lines), and the best fit Gaussian function (dotted line) to each of these distributions. Part A and B are the red and green pixel value distributions of a view that does not contain defects or the stem. Note that the distributions fit well to a Gaussian function. Part C and D contain the red and green color distributions for a second example view of an orange that does not contain defects or a stem. Note that the fit to a Gaussian function is less good, and that the mean and standard deviation of the two distributions are different for this example orange. Part E and F contain the color distributions and best fit Gaussian function for an image that contains a stem and part T and H are the color distributions for a view that contains a defect.

4. Local Defect Search

One approach to the identification of defects is to segment the defect regions from the parts of the surface that are defect free. A detailed analysis of the characteristics of defects leads directly to the observation that there is no regular clustering of defects in color space. Also in several cases, local color variation between pixels from the same orange is not sufficient to classify defects. The multiplicity of defect types has hampered prior attempts to segment the defects from unblemished surface regions, using local structural or textural properties. Furthermore, it is not clear that the computational effort required to identify all possible classes of defects, if it were possible, is justified, since all defect types contribute roughly equally to the final grading decision.

Instead we construct a simple and effective defect detector using a set of masks applied to regions of the image. The extracted local features are passed to a neural network based classifier that has been trained using a large database of samples. In almost all cases, a defect is characterized by a discontinuity in the skin pigmentation. But the color gradient can nevertheless be quite small, and similar to other non-defect sources of variation such as uneven illumination and border effects.

4.1. Defect feature operators

Ordinary edge detectors do not respond strongly to the appearance of surface defects on oranges. Instead, five larger and lower spatial frequency operators were used. In this process each image of the orange is divided into square regions that are the typical size of defects ($N \times N$), and all five operators are applied to both the red and the green color planes of these square regions. Each operator (M_k) is a $N \times N$ matrix, with integer elements that are either one, zero or minus one. The elements (m_{ij}^k) of the first four of these matrices are defined below:

$$\begin{aligned} m_{ij}^1 &= \begin{cases} 1, & i < n/2 \\ -1, & \text{otherwise} \end{cases} & m_{ij}^2 &= \begin{cases} 1, & j < n/2 \\ -1, & \text{otherwise} \end{cases} \\ m_{ij}^3 &= \begin{cases} 1, & i > j \\ 0, & i = j \\ -1, & \text{otherwise} \end{cases} & m_{ij}^4 &= \begin{cases} 1, & i < (N - j) \\ 0, & i = (N - j) \\ -1, & \text{otherwise} \end{cases} \end{aligned}$$

In order to specify the elements of the fifth matrix (M_5) it is useful to define a set of square matrices, with size and with elements constructed in the same way as (M_k) except that $N' = \frac{1}{2}N$. M_5 is composed of four quadrants which are defined to be:

$$m_{ij}^5 = \begin{pmatrix} -m_{ij}^{4'} & -m_{ij}^{3'} \\ m_{ij}^{3'} & m_{ij}^{4'} \end{pmatrix}$$

The neural network based classifier is applied to each of the image regions. The input layer of the neural network contains ten neurons, computed by separately applying each of the five masks to the red and to the green components.

$$I_k^{r,g} = \sum_{i=1}^n \sum_{j=1}^n j = 1 x_{ij}^{r,g} m_{ij}^k$$

The output layer of the neural network contains two neurons, and they are trained to classify defects, and used to estimate the severity of the defect.

The data base used to train this neural network based classifier is the same as the one described above. However in this case the parts of the image that correspond to defects or the stem have been extracted. Figure 4 contains part of the database of defects. Note that there are several examples of the same defect, which differ only in the selection of the start point for the sampled region. Instead of convolving the entire view with the defect detection masks, the computational overhead was reduced by applying the masks to tessellated regions of the image. In this case, the performance of the classification is significantly improved if the neural network algorithm is trained on images that have been translated in both directions.

Figure 5 contains samples of the stem data base. In this part of the analysis the stems are also classified as defects. This data base is then used in the next stage of processing to differentiate between the stems and defects. This stem detection process is only used if the stem has not already been identified in the prior processing of the six views of an orange.

5. Stem Detection

Stem detection is the most difficult step in the process of grading oranges, since the stem region looks quite similar to a defect. In general, global features such as the average color or the distribution histogram of red and green pixel color components, do not show significant differences between stems and defects. The strategy chosen here is to use a neural network based classifier, applied to identify spatial features extracted from the image region that contains the suspected stem.

The stem has a much more regular structure than the defects and has a high degree of radial symmetry, which can be used to aid in the identification

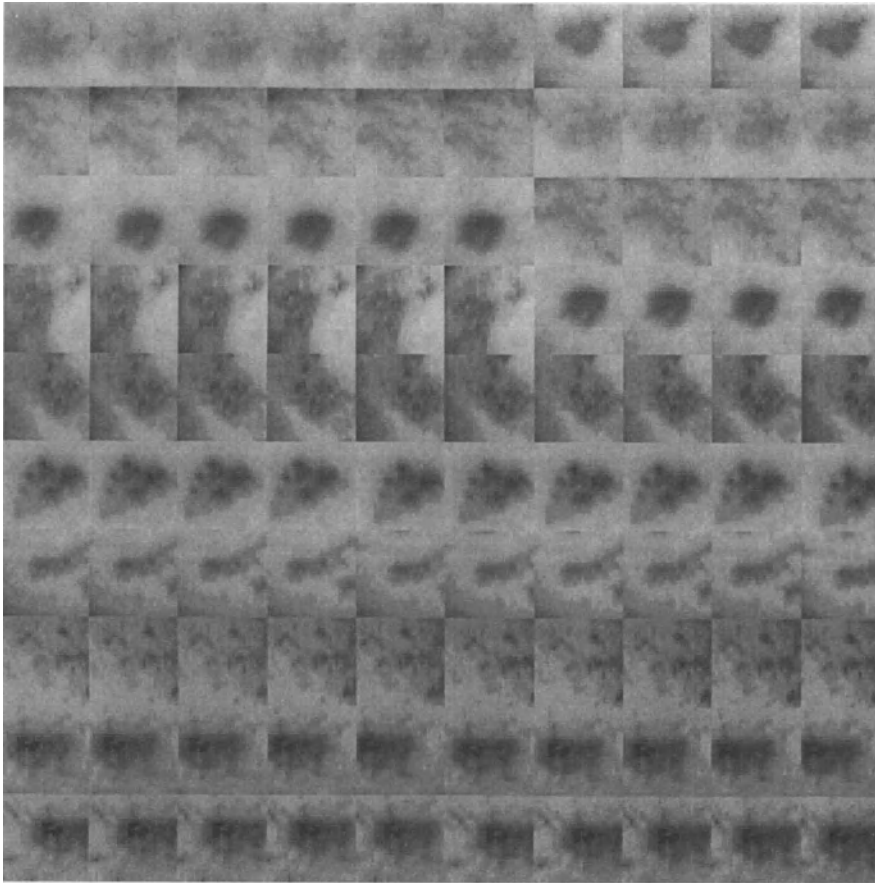


Figure 4. This figure contains 100 examples from the database that is used to train the mask based stem and defect detector. Note that individual defects are entered several times into the database with different displacements from the edge of the mask. The reason for this is explained in the text.

process. Often, but not always, there are characteristic radial grooves that radiate from the stem attachment area. Also in some cases there are concentric rings around the stem attachment area. Unfortunately circular defects, of various types, also occur relatively often. We have found that the family of Zernike moments are very useful in detecting stems. In this process a set of Zernike moment masks are convolved with the region identified by the defect search.

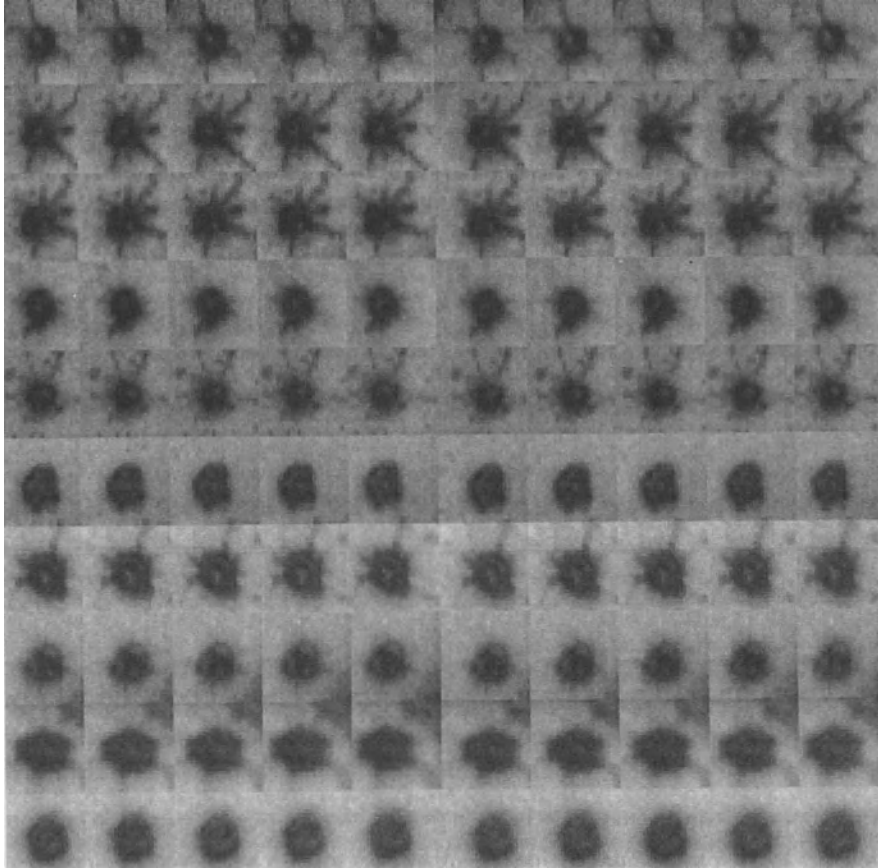


Figure 5. This figure contains 100 examples from the database of stems. This data base is used both to train the mask based defect and stem detection neural network, and is used to evaluate the performance of masks constructed from Zernike moments.

5.1. *Zernike moments*

When used as a convolution mask, the family of Zernike moments is particularly sensitive to circular symmetries. Importantly, Zernike moments are invariant under rotation, which makes this strategy computationally practical. Each of the views of an orange processed by this algorithm has already been scanned for defects using masks much smaller than the Zernike moment mask, so this stage is quite specific to suspected stem regions.

In these tests the radial size of the Zernike moments was not varied, since the size of stems of each particular variety of oranges is relatively fixed.

However, for a commercially viable machine, in which a large range of fruit sizes are processed, some scaling of the mask size may be necessary.

In order to generate a two dimensional square mask based on a Zernike polynomial, the mask is taken to have a unit radius in polar coordinates. Each pixel is assigned a complex value:

$$Z_{nm}(\theta, r) = [\cos(m\theta) + j * \sin(\theta)] * R_{nm}(r)$$

where: n is the major order of the Zernike polynomial, and m is the minor order. The radial variation of the Zernike polynomial results from the term, which is defined by:

$$R_{n,m} = \sum_{s=0}^{s=\frac{(n-m)}{2}} \frac{(-1)^s (n-s)!}{s! (\frac{n+m}{2} - 2)! (\frac{n+m}{2} - 2)!} r^{(n-2s)}$$

which has degree n , with only terms of the same parity as n . It generates an oscillating function with maximum wave number $n/4$. The minor order also affects the radial component, reducing the amplitude of the oscillations in the proximity of the circle center, while enlarging its wavelength. The border oscillations have larger amplitude and shorter wavelength as the value of the minor order approaches the value of the major order.

Figure 6 shows three examples of Zernike masks, represented in the picture with the intensity given by one component. More details on Zernike polynomials and their use as masks for image processing can be found in Khotanzad and Hong (1990).

In these masks, higher frequencies, both in radial and angular directions, vanish inside the mask. This occurs particularly near the center of the mask, where quantization effects are largest. Also the size of the window must be large enough to fully enclose the stem region. Reliable detection is only possible when the mask is well centered with respect to the stem region.

With the Zernike moments, the best input features for a neural network cannot be found empirically. For each Zernike polynomial major order n the possible minor orders are:

$$m = \begin{cases} n/2 + 1, & \text{even} \\ (n+1)/2, & \text{odd} \end{cases}$$

Even if the polynomial order is limited to be less than n_{max} , the number of candidate features (F) is still far too large for exhaustive search. Instead, it was decided to limit the evaluation to a search for the best set over a smaller number of candidate features. Four input neurons were used in this classifier, from 50 candidates required 200,300 runs against the database, which took approximately 100 hours on a SUN Sparc20 workstation.

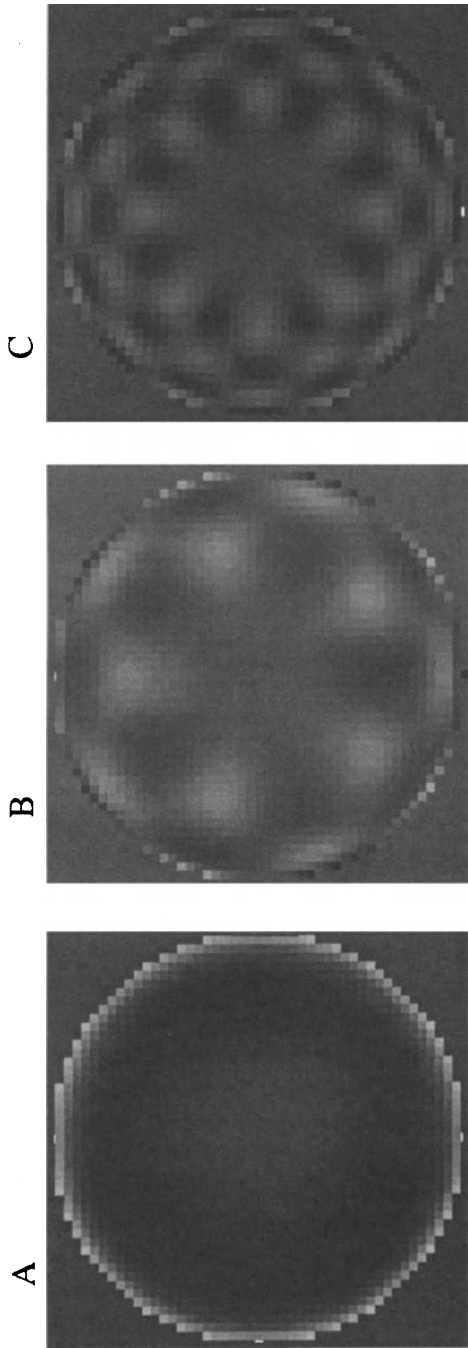


Figure 6. This figure contains three examples of Zernike moment masks. In each case the intensity corresponds to one component of the mask. Part A contains the modulus $Z_{5,3}$ (rank of 38.4 in the red color plane, 44.8 in green), part B contains the imaginary part of $Z_{9,5}$ (rank of 14.8 in the red color plane, 25.1 in the green) and part C contains the real part of $Z_{16,8}$ (rank of 53.3 in the red color plane and 16.8 in the green).

The screening of the subset of candidates was carried out by ranking all the features with a single-variate analysis. For each feature the rank (i) for a feature is computed as:

$$R_i = \frac{|m_{i,s} - m_{i,d}|}{s_{i,s} + s_{i,d}}$$

where

$$m = E[x]; s = E[(x - m)^2]$$

indexed over both the stems (s) and defects (d) in the database.

During the optimization process, ranking is done in the multivariate space. Classification during the analysis and test is accomplished by finding the features that have the smallest average distance to the masks. Figure 6 shows three example Zernike moment masks, and includes the rank of these masks for the red and the green color planes.

6. Results

The work described in this paper is directed towards developing a commercially viable machine with stringent real-time requirements. A successful machine should be cost-effective, robust, simple, and suited for implementation in parallel hardware. Nevertheless, the problem of inspecting fruit is intrinsically complex, and previous attempts to simplify it have been at best partially successful.

The flexible architecture of the algorithm presented here provides an advantage, by splitting the whole task into three simpler problems. The use of neural networks in solving all the steps of the grading has overcome some of the difficulties in defining a reliable description of the properties of the objects to be recognized and classified, which are intrinsically difficult to specify, and the results are so far within the expected performance of the video grading machine.

Figure 7 shows example results of the full grading process on four views of oranges. One of these views contains no defects, two of the views contain defects which have been detected, and the fourth view contains a detected stem.

The initial training and testing of the neural network algorithms was carried out off line using a database of images constructed by running oranges through the machine. The database contains images of 200 oranges of the tarocco variety, and includes 2296 images of stems and 2567 images of defects. Eighty percent of the database was used for training and twenty percent was

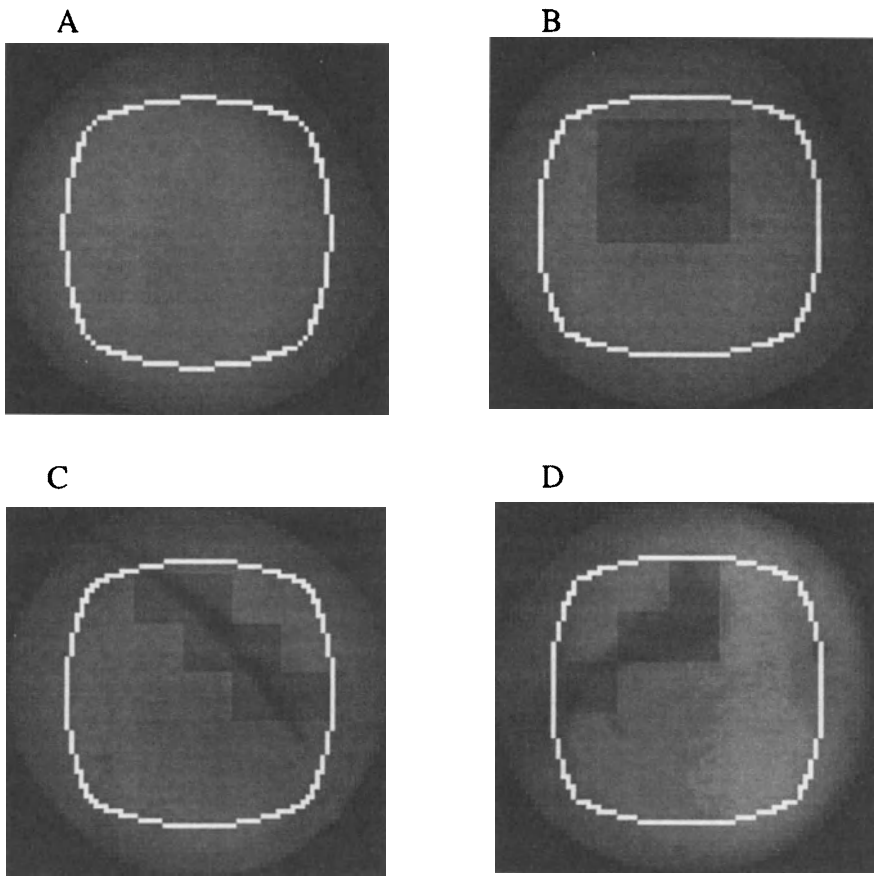


Figure 7. The combined result of applying the full set of processing stages on four example views. The first view (part A) does not contain defects or a stem. The second view (part B) contains a detected stem, and the remaining views (parts C and D) contain views with detected defects. The histogram for these three of the views (A,B,D) were presented in Figure 2 (A,B,E,F,G,H).

used to evaluate the performance of the algorithms. The error rates for each of the algorithmic steps are summarized in Table 1.

The errors in the columns labeled “→ down” represent incorrect down grading of one of the views of an orange. For the histogram evaluation neural network, this means that further analysis is performed on an orange that is defect free. When the neural network used to detect defects erroneously down grades an orange then the region is searched for a stem. If the stem detection down grades the view then the stem is erroneously taken to be a defect. The errors listed in the table in the “→ over” columns are false positives, in which defects and stems are passed undetected.

Table 1. The error rates of the three neural network algorithms. The “→ down” columns are errors that lead to down grading of views of oranges, and the “→ over” column are undetected defects

Network	Training set		Test set	
	→ down	→ over	→ down	→ over
Histogram	0.00%	0.00%	13.73%	2.06%
Defect	0.00%	0.62%	0.80%	2.02%
Stem	0.11%	0.87%	0.44%	1.31%

The imbalance in the two error classes is intentional, as described above. Moreover the differences in performance between the three neural network algorithms reflects the expected relative significance of errors in the final decision. It seems clear that the histogram analysis, which contains no spatial information, is unlikely to be refined significantly. Nevertheless, this is the processing step with the largest difference between the training and test errors, reflecting a rather poor level of generalization. This performance may improve with a larger number of oranges in the training set. For a given number of oranges in the database, the histogram training set is smaller than the data sets for the other two processing stages.

Evidence for the advantages provided by use of neural networks is shown here for the stem detection problem, which is the most important part of the whole algorithm. As a two class problem, the classical Fisher's linear discriminant seems a promising candidate technique (Duda and Hart 1973). When this technique is applied to the sample database, using the same reduced feature number used in the neural network results presented above, the results are quite poor. In this case 27.5% of the stems are not detected and 45.8% of the stems are incorrectly classified as defects.

This linear method may fail due to the implicit assumption that the distribution function of each feature is normal, and this is not true in the case of the Zernike convolutions over the database. In comparison, the neural network algorithm does not explicitly assume the characteristics of the probability distribution of the samples.

7. Summary

The present algorithm for fruit inspection, based on neural network classifiers, has come close to meeting the requirements of a future commercial video grading machine. Further work is envisaged to refine all stages of the classifier, and to balance the computational requirements of each of the stages. Also, thus far, little attempt has been made to use network architec-

tures other than back-propagation learning in a multi-layer perceptron. We plan to compare the performance of alternative non-linear classifiers. The present choice of approach has also been guided by the efficiency of running multi-layer perceptron algorithms on the currently available hardware. The machine has been tested with a few hundred oranges and a range of varieties. However, the machine may require a degree of profiling in order to cope with the large set of varieties of oranges that are commercially available. For example, the machine may need to be adjusted for particularly small or particularly large oranges. Also the machine may require the development of specialized sets of synaptic weights and other parameters in order to optimally grade particular varieties of oranges.

Acknowledgements

This work was partially supported by the European Community CAPRI Project, Sub-project PREFER, of the High Performance Computing and Network Program. The authors would like to thank Nick Ouroussoff and Dave Edwards for invaluable contributions to the design and construction of the video grading machine.

References

- Davenel, A., Guizard, C.H., Labarre, T. & Sevil, F. (1988). Automatic Detection of Surface Defects on Fruit Using a Vision System. *J. Agric. Engng. Res.*: 1–9.
- Duda, R. O. & Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. Wiley-Interscience.
- Grasso, G. & Recce, M. (1996). Scene Analysis for an Orange Picking Robot. In *Proc. of 6th Inter. Congress for Computer Technology in Agriculture*, 275–280. VIAS.
- Hush, D. R. & Horne, B. G. (1993). Progress in Supervised Neural Networks – What's New Since Lippmann? *IEEE Signal Processing*: 8–39.
- Khotanzad, A. & Hong, Y. H. (1990). Invariant Image Recognition by Zernike Moments. *IEEE Trans. Pattern Analysis and Machine Intelligence*: 489–497.
- Maeda, H. (1987). Grade and Colour Sorter Through Image Processing. In *Proc. of Intern. Symp. on Agric. Machinery and Intern. Coop. in High Tech. Era*, 350–356. University of Tokyo.
- Maudit, N., Duranton, M., Gobert, J. & Sirat, J. A. (1992). Lneuro 1.0: A Piece of Hardware Lego for Building Neural Network Systems. *IEEE Trans. Neural Networks*: 414–422.
- Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning Internal Representations by Back-Propagating Errors. *Nature* **323**: 533–536.
- Shearer, S. A. & Payne, F.A. (1988). Color and Defect Sorting of Bell Peppers Using a Machine Vision System. *Trans. ASAE*: 1–9.
- Tillett, R.D. (1991). Image Analysis for Agricultural Processes: A Review of Potential Opportunities. *J. Agric. Engng Res.*: 247–258.
- Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University.

Cell Migration Analysis After *In Vitro* Wounding Injury with a Multi-Agent Approach

ALAIN BOUCHER¹, ANNE DOISY², XAVIER RONOT² and
CATHERINE GARBAY¹

¹Laboratoire TIMC/IMAG, Grenoble, France; ²DyOGen (UPRES n° EA 2021), INSERM U309, Grenoble, France and Laboratoire de Neurobiologie du Développement, E.P.H.E., Montpellier, France (Authors can be contacted at: Institut Albert Bonniot, Domaine de la Merci, 38706 La Tronche Cedex, France; E-mail: Alain.Boucher@imag.fr)

Abstract. This paper presents a multiagent system for studying *in vitro* cell motion. A typical application on the wound closure process is presented to illustrate the possibilities of the system, where different image sequences will be treated. The motion issue involves three aspects: image segmentation, object tracking and motion analysis. The current system version focuses mainly on the image segmentation aspect. A general agent model has been designed, which will be further expanded to include tracking and motion analysis behaviors as well. The agents integrate three basic behaviors: perception, interaction and reproduction. The perception evaluates pixels upon static and motion-based criteria. The interaction behavior allows two agents to merge or to negotiate parts of regions. The negotiation can be seen as a segmentation refinement process done by the agents. Finally, the reproduction behavior defines an exploration strategy of the images. Agents can start other agents around them, or they can duplicate themselves in the next frame. The frames are processed in pipeline, where previous information is used to treat the current frame. One unique agent model exist. Agents are specialized on execution time according to their goals. The results, coming from an existing prototype, show different types of cell behavior during cell migration, based on cell nuclei analysis.

Key words: cell migration, computer vision, multi-agent, wound healing

1. Introduction

The study of cell motion and behavior is important to the understanding of the mechanisms involved in areas such as wound healing, host defense behavior and propagation of diseases. Quantitative analysis of the cellular deformation and motion appears to be an essential tool to improve knowledge on the cell's migratory properties. In this paper, we focused our attention on a particular issue of the migration problem, the wound process.

To study this phenomena, we designed a multi-agent system. The study of motion includes three aspects: image segmentation, object tracking and motion analysis. The main aspect developed in this system is the image

segmentation. The other two are linked with the segmentation process. This process uses motion information to perform its task and can help in the further tasks of motion tracking and analysis. A general agent model has been designed, which will be further expanded to include specific tracking and motion analysis behaviors as well.

1.1 The cell wound process

Cell migration has been described as a major event of tissue formation and remodelling which takes place during several physiological processes, such as embryonic morphogenesis and wound healing. Cell migration is also an important component of some pathological situations such as invasion and metastasis of tumor cells (Mareel et al. 1990). Cell motility can be explained by cell deformations and cytoplasmic forces. These constraints can be transmitted to the extracellular substrate by means of transmembrane receptors like integrins (Sims et al. 1992). This constraint field varies with intracellular and extracellular stimulations. Thus, cell migration has been described as an increasing concentration of a soluble factor (chemotaxy) or along a concentration gradient of a substratum component (haptotaxy) leading to the modification of the cell response (Lester and McCarthy 1992; Bernstein et al. 1993). There are various methods to study cell migration such as the Boyden chamber technique (Dunlevy and Couchman 1993), *under agarose* assays (Stokes et al. 1990; Muller et al. 1993) and *in vitro* wound assays (Matthay et al. 1993; Watanabe et al. 1994). This approach is of interest for investigating *in vivo* wound healing mechanisms: for instance the presence of macrophages has been shown to be required for the regeneration of many cell types during wound healing (Rappolee et al. 1988) and microvascular endothelial cells seem to play a central role in wound healing (Karasek 1989). Cell migration during wound healing could also be influenced by extracellular matrix factors (Person et al. 1988). This method can be ideally combined with *in situ* hybridization procedures thus allowing a direct comparison of the gene expression pattern of the migrating cells with that of the stationary ones (Scharffetter et al. 1989). Drug effects on this process can also be assessed by studying rate of wound closure (Wong et al. 1994; van Luyn et al. 1992; Bodor et al. 1991; Smoot et al. 1991). Moreover, wounding has been demonstrated to be critically involved in tumor formation, leading to the notion that tumors can be regarded as wounds that never heal (Schuh et al. 1990). Thus, it is important to understand whether the induction and regulation of cell motility during wound repair differ from those involved in tumor cell motility (Matthay et al. 1993).

However, tools available for analysing cell migration results are not yet very powerful. On the one hand, using macroscopic methods such as the

Boyden chamber assay, it is possible to evaluate cell mean density. But this approach is rather long and poorly reproducible. Furthermore it precludes spatial and temporal analysis of cell dynamics. On the other hand, microscopic techniques usually use fixed cells.

In this context, computer-assisted image analysis is very useful for following up cell motion. Without perturbing its functions, this method makes it possible to investigate the chromatin organization, which has been shown to play a central role in tumor evolution (Palcic 1994; Beil et al. 1995; Francisco et al. 1995). However, microscopic techniques usually require fixed cells. On the other hand, DNA quantification is possible using the fluorescent non perturbing staining with the vital bisbenzimidazole probe, Hoechst 33342 (Paillason et al. 1995). Moreover, in order to minimize Hoechst effects on the cell cycle, verapamil, a calcium entry blocker, has also been added (Leitner et al. 1995).

1.2 Computing cell motion

Three main aspects are known to be associated with the motion problem: image segmentation, object tracking and motion analysis. These aspects will be considered at different levels.

Most research on visual motion aims at either one or another of these aspects. But few papers exist on combining these techniques. Some work has been done on combining detection and object tracking (Parvin et al. 1995). In our case, we simplify the object tracking issue (Naudet et al. 1996). We do not consider occlusions nor excessive displacements of objects. This is justified by the cell motion observed in the images. Cells move relatively slowly and on a planar slide. The system follows objects by keeping links between the images during the segmentation process. Migration can be easily represented in this way.

Different approaches have already been suggested for motion analysis. Motion is often retrieved from pixel correspondence (Schweitzer 1995), or optical flow (Siegert et al. 1994). The image motion is computed from constraints derived from pixels and from models. We use motion as part of the object detection and tracking to guide the image segmentation. It can be further analysed through some statistics collected during the segmentation process.

The hardest problem is the cell segmentation, the quality of which strongly conditions the tracking and analysis steps. For a long time, this problem has been seen as applying edge or region detectors to an image, especially in cytology (Fu and Mui 1981). We now assist at two successive progress lines. First, the use of *a priori* knowledge (Liedtke et al. 1987) and models allows us to cope with the difficulty of moving image segmentation. Secondly,

most of the research concluded that there was a necessity of a cooperation between methods, and to their adaptation to the local situations in the image (Cocquerez and Philipp 1995).

These conclusions lead to the consideration of a multi-agent system. Behavior-based systems have received much attention for the past few years, and have given rise to different integration models (Maes 1989; Steels 1990; Brooks 1991) which integrate the abilities of the separate behaviors. In the segmentation domain, knowledge-based systems have been developed, where agents use problem-related knowledge (Baujard and Garbay 1993; Boissier et al. 1994), to adapt to application needs. Some reactive approaches, like Bellet's cooperative system (Bellet et al. 1995), were also studied, where edge and region growing methods co-operate in a multi-process data-driven system for low level image segmentation.

The segmentation of cell images has been considered by many researchers and has been a difficult problem (Liedtke et al. 1987; Wu et al. 1995; Cloppet-Oliva and Stamon 1996), due to the contrast quality and to the deformable structure of the cell. With respect to the first point, the contrast intensity between the cells and the background is low, as is shown by the unimodal histogram of the images. The intensity of the background is nonuniform across the image, due to light source in microscopy. High local variation of intensities is observed both within the cell and the background. To attempt to solve these difficulties, several methods have been tried, like active contours methods (Leymarie and Levine 1993; Gwydir et al. 1994; Leitner et al. 1995). Even nucleus detection is difficult (Lockett and Herman 1994). In our application, cell segmentation and tracking is aided by the introduction of a second image showing the nuclei acquired in fluorescence.

With respect to the second point, cell motion introduces some other problems, because cells are non-rigid and deformable. Over-constrained global descriptions of shape and motion are a kind of approach used mostly for human motion (Leung and Yang 1995), but for cells there is no existing model. We preferred differential techniques, because of their adaptability and locality. Motion information is provided using a differential operator (Jain 1981) used to estimate the motion between frames. Motion is integrated in the segmentation process, and this simplifies further motion analysis and object tracking.

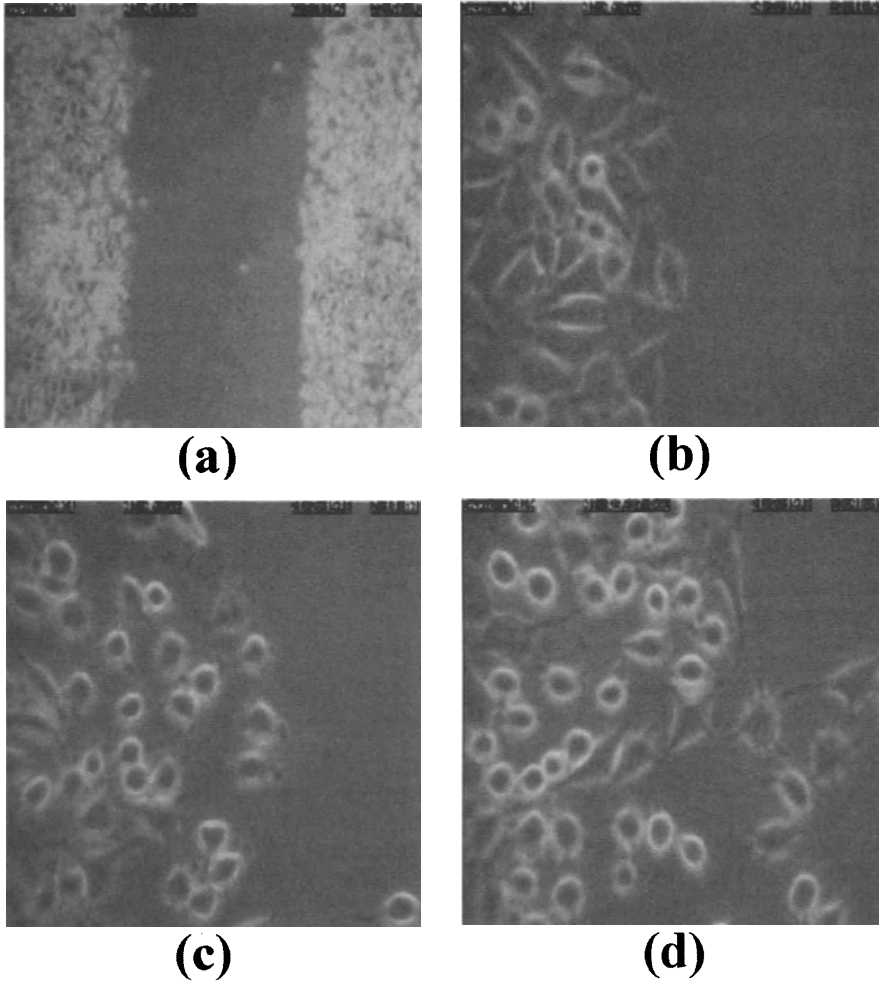


Figure 1. The wound process. (a) Example of a typical injury in a fibroblast monolayer, (b) (c) (d) Time course migration of cells during wound closure. (b) $t = 0$. (c) $t = 13\text{h}$. (d) $t = 24\text{h}$.

2. Materials and Methods

2.1 *Biological experimental procedures*

2.1.1 *Cell culture*

Anchorage dependent L929 murine fibroblasts (ATCC, CCL 1) were grown to plateau phase in antibiotic free Minimum Essential Medium (MEM, Techgen) supplemented with 5% fetal bovine serum (FBS) and L-glutamine. Cultured cells were verified to be free from mycoplasma contamination.

2.1.2 DNA staining

Verapamil (VER) and Hoechst 33342 (Ho 342) were purchased from Sigma. Stock solutions of VER and Ho 342 were prepared in water, respectively at a concentration of 10^{-2} M and 10^{-3} M. Cultured cells were then washed in Phosphate Buffered Saline (PBS) and incubated with serum-deprived MEM medium for 30 min, VER and Ho 342 were added in MEM medium (final concentrations 40 and 4 μ M respectively) and cells incubated for 30 min. They were washed three times with fresh serum-supplemented MEM medium.

2.1.3 Artificial wounding

Three days after cell culture in glass chamber (Lab-Tech), L929 cells form a confluent monolayer in culture. This confluent cell monolayer is then gently scratched with a Gilson pipette blue tip without damaging the glass surface and then thoroughly rinsed with medium to remove all cellular debris (Figure 1a). Cell migration during the artificial wound closure was computed using microscopic imaging.

2.1.4 Microscopic imaging

The image sequences were obtained using an imaging workstation, composed of an inverted microscope (Axiovert 135M, Zeiss) equipped with a temperature and CO₂ controlled chamber and a SIT camera (C2400-08, Hamamatsu) connected to an analyser (SAMBA TM 2640, Unilog). The modification of cells position was observed at regular intervals for several hours (Figures 1b, c and d). Both phase contrast (cells) and fluorescent Ho 342 labeled (nuclei) images were registered (Figure 2). The images of fluorescent nuclei were acquired using a combination of an excitation-filter passing wavelength 365 nm, a dichroic filter passing 395 nm and an emission high-pass filter for wavelength 450 nm (when bound to λ emission and excitation of Ho 342 where 340 and 450 nm respectively).

2.2 The multi-agent system description

A multi-agent system has been designed to assist the analysis and interpretation of cell migration as previously presented. The present design is mainly oriented toward image segmentation. Cell tracking and motion analysis are currently integrated as part of the segmentation process. A general agent model has been designed, which will be further expanded to include complete tracking and motion analysis capabilities. Our segmentation of cell images distinguishes four different zones (Figure 2):

1. **Nuclei** are the cell's hearts. They are textured and therefore hard to delineate, but they are easier to find with the fluorescence image.

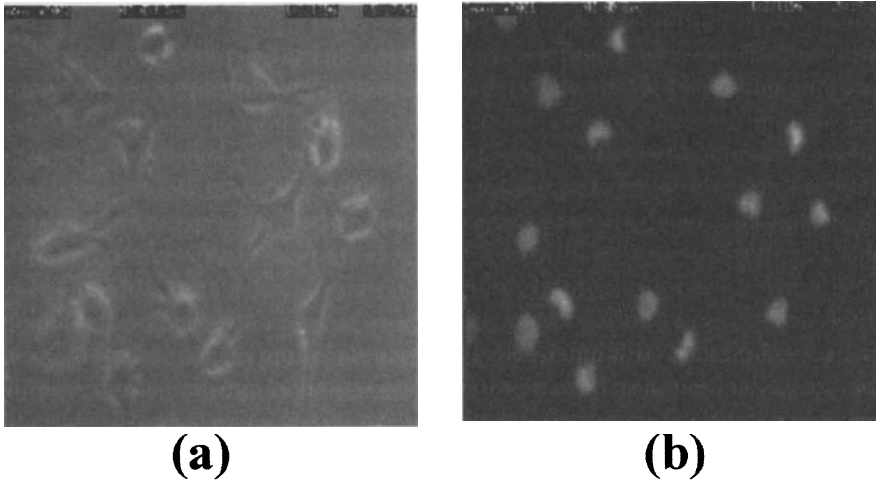


Figure 2. Images of fibroblasts L929. (a) Image of cells observed in phase contrast. (b) Image of nuclei observed by fluorescence.

2. **White halos** circle the cell. They result from light diffraction around the cells. They look like halos which blur the cell's exact boundary.
3. **Pseudopods** are cytoplasmic extensions of the cell. They represent the hardest part of the segmentation problem. They are poorly contrasted from the background, but their motion is important. We segment the cytoplasm around the nucleus along with the pseudopods.
4. **Background** is segmented to define the cell's boundary. Its intensity is nonuniform across the image, because of the light source used in the microscope.

Segmentation is modelled as a complex task combining perception, merging, negotiation and reproduction. Each subtask is modelled as a specific behavior of the agent. The behaviors are controlled by an internal manager. The agent society is supervised by a scheduler, according to the specification of the external user. The system is divided into several levels (Figure 3):

- The **user** interacts with the system through the sequencer. He can create or delete agents, or modify their priorities on execution time. He can also configure each agent before execution through a dedicated panel.
- The **sequencer** manages the agent's execution. It is similar to an operating system. Its goal is to manage all the agents into the same UNIX process.
- The **internal manager** controls the selection and the execution of the agent's behaviors. A mechanism of priorities driven by events allows the choice of a behavior for execution.

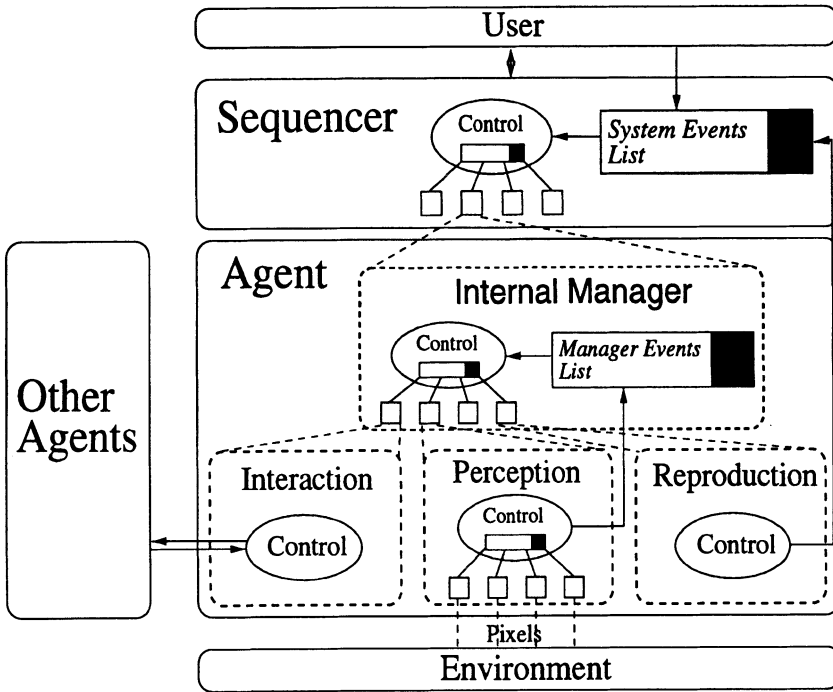


Figure 3. Multi-agent system model.

- The agent **behaviors** perform the segmentation and control the exploration of the images. They are three kinds: *perception* (normal / focalized region growing), *interaction* (merging / negotiation) and *reproduction* (into the same frame / toward the next frame).
- The **environment** includes the input images, their characteristics, and the results.

There is only one agent model, specialized at execution time. Each agent works on one region of a particular frame. For example, a nucleus agent works on the segmentation of one nucleus in one particular frame of the sequence.

Figure 4 shows the system state while working on an image sequence. Each agent's goal is to find a given component in a given frame. Segmentation of all the frames is done through a pipeline. When the current segmentation state of a frame is sufficiently advanced, agents are created in the next frame. Segmentation results of a frame can be accessed by all the agents working on this frame and by those working on the next one (information and results of a frame i is available for all agents working on frames i and $i + 1$).

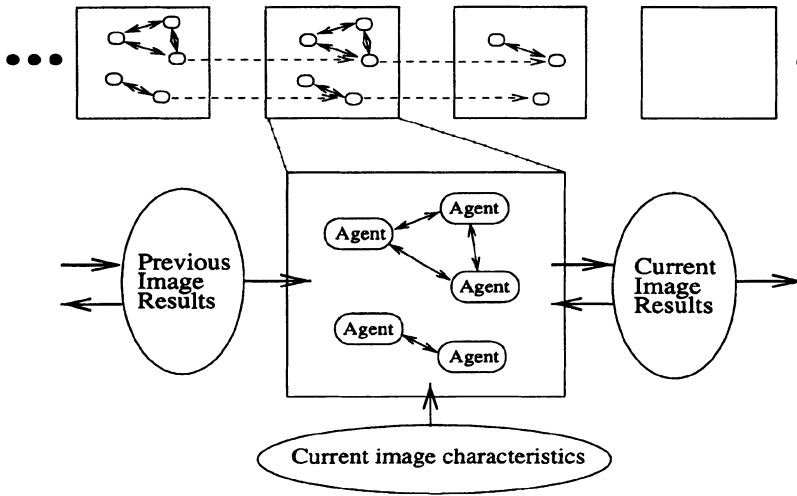


Figure 4. Segmentation of an image sequence: dynamic creation of agents and results transmission. An agent in one frame keep a link with its successor in the previous frame.

2.2.1 Sequencer

Our system is designed as one UNIX process. Thus, we create a sequencer, similar to an internal operating system. Two main reasons explain this choice:

1. The number of agents in the system is very high. There could be at the same time as many as ten agents for one cell in one frame. Due to this, the maximum number of allowed processes in a computer is rapidly reached.
2. This approach gives us the flexibility to control the system's simulation parameters.

Agents are chosen to be executed one by one. The execution time is used by the agent as it needs, but at the end of the allowed time the agent is responsible for returning the control to the sequencer. Agent execution is influenced by external events. These events are requests from the user or from the agent itself. Two kinds of events exist: create or delete an agent.

2.2.2 Internal manager

The internal manager ensures the relation between the agent with the sequencer. It manages the agent behaviors and it is implemented as a control loop. It selects the most appropriate behavior, with the highest priority, executes it and then adjusts the priorities of all other behaviors. These behaviors produce events during their execution. Events are stored in a list by the manager, and priorities are adjusted according to the accumulated events (Table 1). The

Table 1. Internal manager events and involved behaviors. An event (in first column) is sent to the internal manager by a behavior (in second column) during its execution and will affect (increasing or decreasing) the execution priority of another behavior (in third column).

Events	Origin	Modified priority behavior
Pixel with same region type added	Normal growing	Fusion (request)
Pixel with different region type added	Normal growing	Focalized growing
Focalized growing finished	Focalized growing	Negotiation (request)
Merging / negotiation requested	External agent	Fusion/negotiation (answer)
Merging / negotiation answered	External agent	Fusion/negotiation (action)
Size of region increased	Normal growing	Reproduction (next frame)
Normal growing finished	Normal growing	Reproduction (same frame)

way priorities are defined and computed has been adjusted experimentally. Further work will provide a more accurate handling of the parameters.

At each loop, the behavior with the highest priority is executed. Its execution time depends on the time conceded by the sequencer to the internal manager. This time can be managed between the behaviors as needed, as long as the manager gives back the control to the sequencer when it is elapsed.

As a consequence, the agent scheduling is not fixed in advance. It is updated during execution according to incoming events. The agent basic behavior is normal region growing and is always present. Its termination means the end of the agent's life, since its primary goal is segmentation. Other behaviors are activated depending on the events produced by this first behavior, and will produce other events themselves. They can also react to events from other agents.

2.2.3 Behaviors of agent

Three different kinds of behavior exist: *perception* (normal / focalized region growing), *interaction* (merging / negotiation) and *reproduction* (into the same frame / toward the next frame). The perception and reproduction behaviors can be configured at execution time depending on the segmentation type the agent has to perform (nucleus, pseudopod, white halo or background).

Perception: Normal growing.

Normal growing is the basic behavior. It is responsible for analyzing the local environment, evaluating it, and elaborating a list of candidate pixels. The region growing method is based from the work of Bellet (Bellet et al. 1995) and depends on candidate pixels. At each loop, the pixel with the highest evaluation is chosen from the list and added to the region. Its neighbors are then evaluated (2 adjacent neighbors in the horizontal direction and 2 adjacent neighbors in the vertical direction). An evaluation function is used to order

the candidate pixels. This function weighs various static and motion-based criteria. The general expression of this function is:

$$\varepsilon = (1 - \gamma)\varepsilon_{static} + \gamma\varepsilon_{motion}$$

$$\varepsilon = (1 - \gamma) \sum_{i=1}^6 \alpha_{s_i} e_{s_i} + \gamma \sum_{i=1}^3 \beta_{m_i} e_{m_i}$$

where γ is the motion evaluation weight in the global evaluation, α_{s_i} and β_{m_i} are the weights of the static and motion-based criteria respectively, e_{s_i} and e_{m_i} are the evaluations of the respective criteria. All these values can vary from 0 to 1.

Six static criteria are used: variance similarity, compactness, gray level similarity, gradient direction similarity, cell and nucleus image thresholding. The nucleus image thresholding method is evolutive and adaptative. It is computed on the gray level (according to the nucleus image) of the region and updated at each newly aggregated pixel (*threshold = mean gray level $\times$ factor*). This mean gray level changes along with the segmentation process, as the threshold does. Therefore, it is computed dynamically as the situation evolved. All other criteria are defined in a similar way. An example with the variance level criterion is:

$$e_{variance} = \begin{cases} 1.0 & \text{if } 0 \leq \text{variance} \leq v_1 \\ 0.6 & \text{if } v_1 < \text{variance} \leq v_2 \\ 0.4 & \text{if } v_2 < \text{variance} \leq v_3 \\ 0. & \text{else} \end{cases}$$

v_1 , v_2 and v_3 are computed automatically from the variance histogram. From the usual static criteria, we developed other criteria based on motion information. We use two motion information sources: the difference image, computed between two following frames, and the segmentation of the previous frame. Four different motion-based criteria have been developed. Figure 5 shows two examples.

The criteria weights, the static / motion weight (γ), and the thresholding factor for aggregation are specialized for the type of object to be segmented (nucleus, pseudopod, white halo, background). Pixel evaluations are then different and adapted to the particular objectives of the agent. These parameters are given fixed values.

All the neighboring pixels are evaluated and added to the candidate list. If one chosen pixel is already labelled by another region, then it is added to the manager's event list. In so doing, we take note that a segmentation error might have occurred, or that two regions might describe the same entity. These

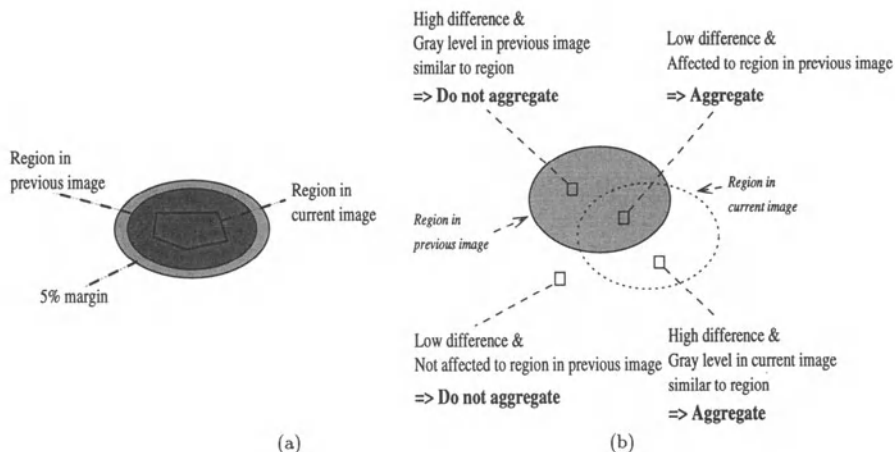


Figure 5. Motion criteria for evaluation of candidate pixels. (a) Region in the current frame can not grow more than its previous size more or less a 5% margin. (b) If difference for a pixel is high, gray levels are compared to find in which frame the pixel belongs to the region. If difference is low, the pixel is aggregated as in the previous frame.

pixels will be further examined and behaviors of focalized growing, merging or negotiation will be launched if necessary.

Motion statistics are computed by the normal growing behavior. The distance and direction of the motion are known when the region is completely segmented from the previous segmentation of the same region.

Perception: Focalized growing.

The normal growing behavior does not exclude any pixel, and may even consider some pixels that have been already labelled by another agent. This means that these pixels may be consistently evaluated by two different agents. If the two agents types are the same (two background agents for example), then the two regions can merge. If they are of different types, they will have to negotiate the zone of common interest because of the resulting ambiguity. Before doing this, the zone to be negotiated has to be defined. It is the objective of the focalized growing behavior.

This behavior focalizes the agent's attention on some part of the image already labeled by another agent. It is launched when enough pixels of interest are found. Its objective is to determine which and how many pixels are involved in this situation. The aggregation threshold is raised to avoid too much dispute between the agents. The ultimate goal of the negotiation behavior is to correct the segmentation that has been currently obtained.

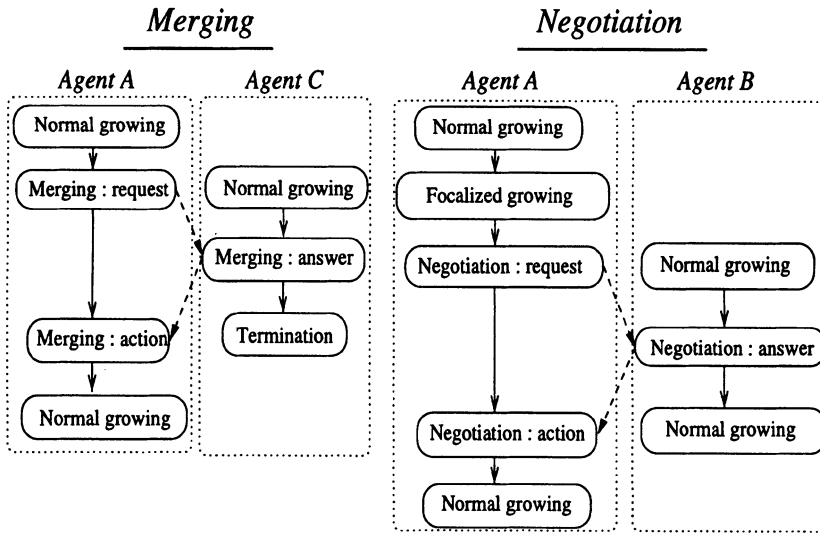


Figure 6. Interaction processes.

Interaction: Merging.

The merging behavior permits the fusion of two regions describing the same entity. It is launched when some pixels in the manager's event list show that a neighboring region has the same type and similar characteristics. It will then communicate with the other agent and request the merging of the two regions. One of the two agents will then terminate.

This behavior is activated only if there are enough pixels accumulated in the manager's event list. There are three steps which structure the dialogue between the two agents (Figure 6). An example is shown in Figure 7. Agent A communicates with agent C and requests a merging. Then, agent C analyses the situation and communicates its answer. As a result, the regions will possibly merge, and if they do, agent C will terminate itself.

Interaction: Negotiation.

The negotiation behavior concerns two regions of different types. It allows two agents to go into conflict for a zone and to negotiate it. The conflicting zone is obtained by a focalized growing behavior performed by one agent, and by a normal growing behavior performed by another. The objective is to proceed to some segmentation correction. For this purpose, the two agents must evaluate the conflicting zone and compare their evaluations.

There are three steps to accomplish this task (Figure 6). An example is shown in Figure 7. Suppose that agent A has performed a focalized growing over some part of region B. Then, it communicates with agent B, sending

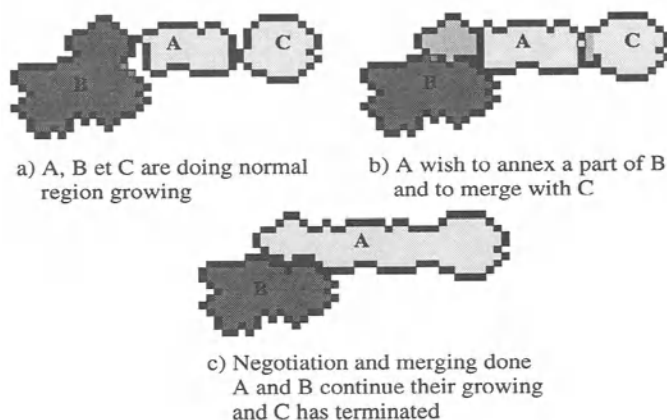


Figure 7. Interaction between agents.

its personal evaluation of the zone. Agent *B* re-evaluates the zone itself and compares both evaluations. Depending on the comparison, the zone will be annexed by *A* or kept by *B*. It will then be declared inviolable to avoid multiple disputes over one part of the image. The negotiation behavior is activated only if the focalized growing behavior has found a sufficiently large zone that is worth launching the negotiation process.

The main difference for launching the merging or the negotiation behavior is in the type of the regions involved. Two agents describing regions of the same type will merge, but two regions describing regions of different types will negotiate some parts of themselves. The priorities of both behaviors are adjusted according to the number of pixels involved.

From their results, the two behaviors may be very different. Merging means that two agents put their work in common. In the negotiation behavior, an agent questions the work done by another agent, and from their conflict, segmentation correction and improvement may be done.

Reproduction: Into The Same Image.

The reproduction behavior allows an agent to initialize other agents, of all types, around its current position. Two events can activate this behavior:

- the **beginning** of the agent's life: the newly created agents will then compete with their creator and a natural selection for the pixels occurs. Agents are thus working in parallel and in competition.
- the **end** of the agent's life, corresponding to the end of its normal growing behavior: the newly created agents will benefit from the work already done. Agents are then launched in a more predictive way. We say they are working in serial and under planned cooperation (easiest work first ...).

This behavior tries to maximize co-operation and result-sharing to ensure the achievement of the global goal, the segmentation of the whole image. Examples of reproduction are shown with the results (Section 3). Reproduction is specialized, like the perception behavior. The specialization includes the type, number, time, and place where new agents will be created.

Reproduction: Toward The Next Image.

The duplication behavior allows agents to initiate the segmentation of other images. When its segmentation is judged sufficiently mature (bigger than a fixed size), an agent can initiate the segmentation of that region in the next image by duplicating itself. In this way, the segmentation of all the frames is done through a pipeline. An agent working on one frame can, furthermore, access the previous image information.

The maturity of a region is computed from its size (depending on the region type). The more a region grows, the more its behavior's priority increases. When sufficiently high, the behavior launches one identical agent at the same place in the next frame. A specific link is created between the newly created agent and its initiator. Any agent knows which one launched it and can thus retrace its motion from this knowledge. The motion segmentation is based on links kept between agents on different frames.

By the two different mechanisms of reproduction, we can define a segmentation strategy adapted to the application constraints and dynamically elaborated, based on local rules. Indeed, the sequence exploration strategy is not predefined. It emerges from the agent activation and findings.

3. Results

The results shown in this section are generated by a prototype which can apply the normal growing, merging and reproduction behaviors. Each agent in this prototype is conceived as a separate UNIX process, so that there is no sequencer nor internal manager. The specialization of behaviors comes from a configuration file.

This section illustrates 4 different kinds of results: (i) the user interface for configuration and visualization, (ii) discussion of segmentation results on some frames, (iii) cell tracking results for random motion and for the wound process are compared, and (iv) the different cell motions are analyzed.

3.1 User Interface

A user interface has been developed for better use of the system. This interface relies on two components. First, a configuration file allows the user to specify

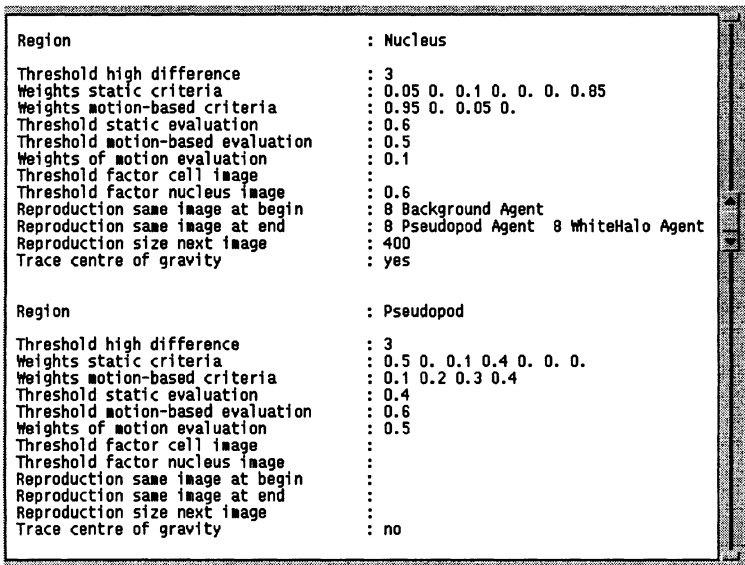


Figure 8. System configuration interface.

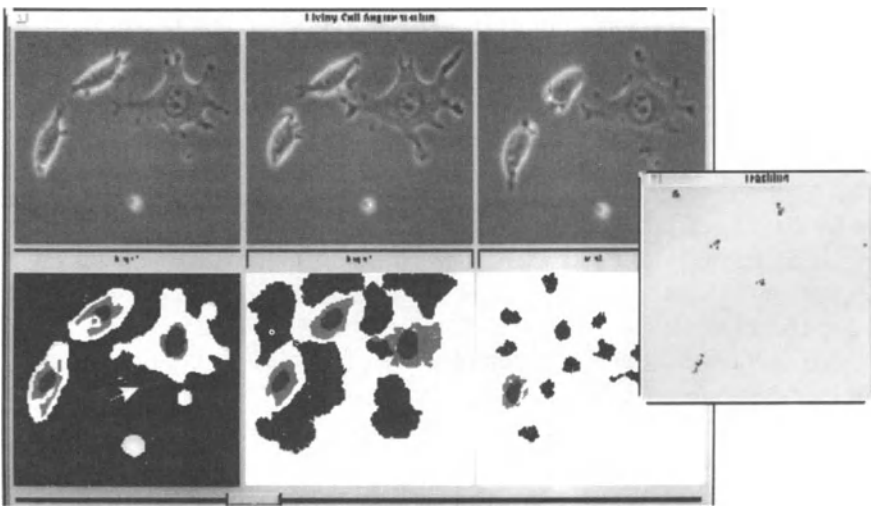


Figure 9. System user interface. The left window shows the segmentation process. The upper images are the original cell frames, and the lower, the results of segmentation. The right window shows the nucleus tracking in the sequence.

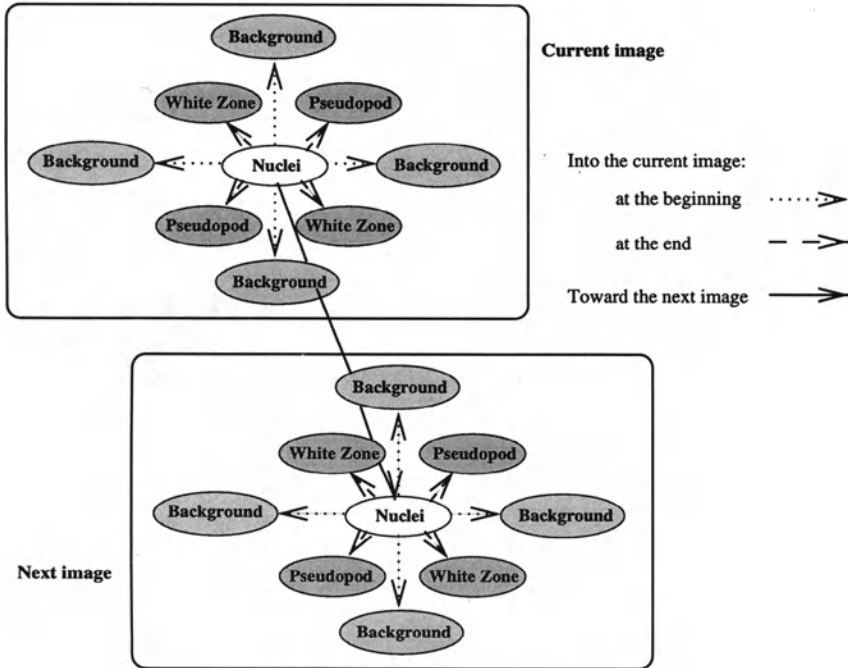


Figure 10. System exploration strategy: a local plan is defined at the nucleus agent level.

the execution parameters of the agents and through this, to create specialized agents (Figure 8). This specialization comes only from this configuration. By naming agents and defining interaction between them, the user adapts the system to the specific application.

The second component of this interface is the output window of the image segmentation (Figure 9). The sequence is segmented in pipeline. The user can see the dynamic of the system through the sequence of images. The higher images are the cell frames and below, the segmentation results are shown during computation. A browser allows the user to move in the sequence. The trace window on the right plots the center of gravity of all the nuclei founded in the sequence.

3.2 Segmentation

As previously explained, the reproduction behavior allows us to define an organization strategy adapted to the application. Figure 10 presents the actual exploration strategy in our system. Nucleus agents are launched first, based on seeds computed automatically from two parameters the user can configure: a maximum number of seeds and a minimum distance between seeds.

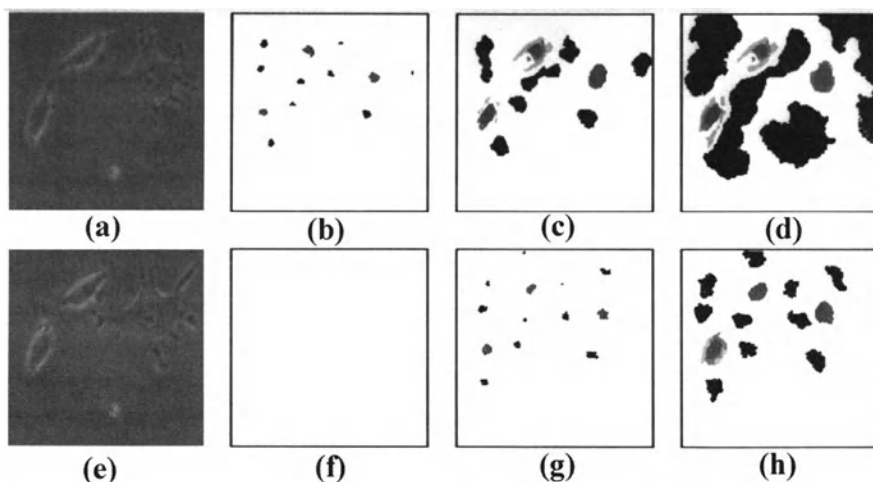


Figure 11. System evolution. (a) Cell frame 1. (b) (c) (d) Evolution of segmentation for the frame 1. (e) Cell frame 2. (f) (g) (h) Evolution of segmentation for the frame 2. The second frame is segmented when the first one is sufficiently segmented. The sequence segmentation is done through a pipeline. Time interval between frames is 15 min.

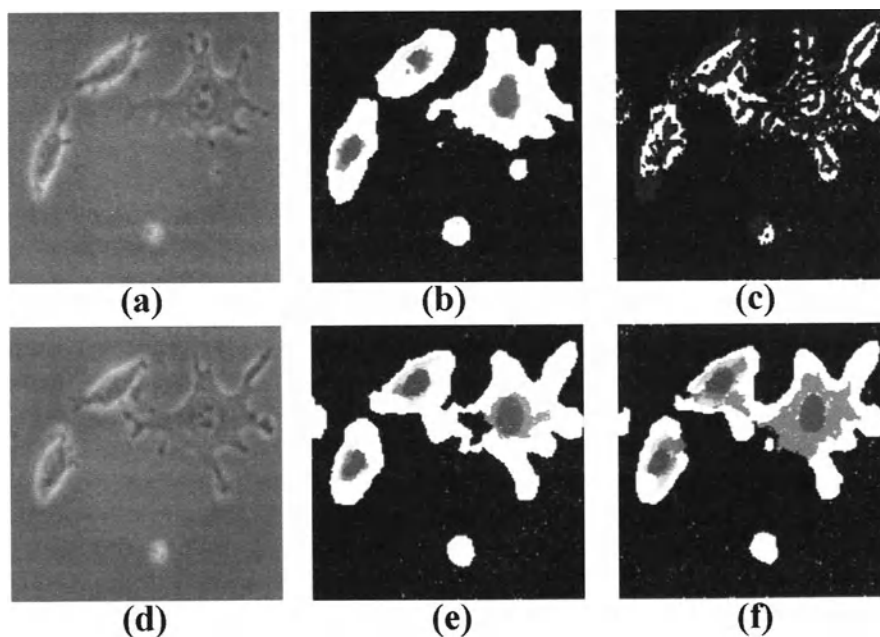


Figure 12. Segmentation using static and motion-based criteria. (a) Cell frame 1. (b) Segmentation of frame 1. (c) Difference image between frames 2 and 1. (d) Cell frame 2. (e) Segmentation of frame 2 using only static criteria. (f) Segmentation of frame 2 using static and motion-based criteria. Time interval between frames is 30 min.

Indeed, the nucleus agents are launched at the brightest pixels, according to the nucleus image in fluorescence. Moreover, a minimum distance must be respected between seeds to avoid having all the agents started on the same nucleus.

At the outset, nucleus agents launch background agent seeds surrounding them because background takes longer to segment and because information from its segmentation is useful to later segment pseudopods and white halos. Nucleus agents in the subsequent frames are started when the current frame segmentation is judged sufficiently advanced. When the nucleus segmentations are finished, pseudopod and white halo agents are finally started around the existing nuclei. Their development is constrained by the nuclei and the background. Figure 11 shows the system evolution. The agents are launched in the first frame (Figure 11b). When the segmentation of this first frame has sufficiently progressed, new agents are automatically launched in the second frame (Figure 11g). The segmentation of the two frames are completed in parallel (Figures 11d and h).

The number and the kind of agent launched can be configured at execution time. When starting the system, the user specifies the (maximum) number of nucleus agents to be initialized. Other launchings are configured in the configuration file. This file indicates the number and kind of agents to be launched at the beginning and at the end of the agent's life.

The perception behavior evaluates pixels with two different series of criteria, relying on static or motion-based information, except for the first frame, which is segmented only with static information. Figure 12 shows the segmentation results for two successive frames in a sequence. Frame 1 (Figure 12a) is segmented with static criteria (Figure 12b). Frame 2 (Figure 12d) is segmented with static criteria only (Figure 12e) and with static and motion-based criteria (Figure 12f) to illustrate the use of both kind of criteria. The difference image between both frames is shown in Figure 12c. The background and the nuclei are generally well segmented. The frontiers and the center of the cells are well defined by these regions. Between them, the pseudopods and the white halos show variable results. The use of motion-based criteria can influence largely the results of these zones. The difference image give additional information to help defining the exact positions of the pseudopods and the white halos. These preliminary results encourage us to continue our effort in the refinement of the motion-based segmentation criteria.

The complete segmentation of an image takes about 5 minutes on a Sun Sparc 5. But the average time on a sequence is less because the images are segmented in pipeline.

3.3 Object tracking

At the end of an agent's life, a trace of its center of gravity is plotted in the trace window. Lines are also plotted to link one component in two successive frames, and therefore to show the displacement of the component between these frames. In the current version of the system, only nucleus agent traces are usable, due to the difficulty of segmentation of the other components. As an extension of this work, we want to link the different cell zones and study the cell behavior when moving.

Four sequences were studied, and will be referenced with the following names:

- Natural motion: two different image sequences
 - 19 frames every 30 min (NM30);
 - 20 frames every 3 min (NM3).
- Wound closure: two image sequences representing two phases of the same process
 - 27 frames every 30 min (first phase: $t = 0$ to $t = 13h$) (WC1);
 - 15 frames every 30 min (second phase: $t = 17h$ to $t = 24h$) (WC2).

Figure 13 shows the tracking results of the cell's nuclei for the 4 sequences. The lines shows the nuclei that the system has been able to track. Single dots indicate the nuclei that has been found in one frame only, or nuclei that could not be linked with other frames. Tracking of some nuclei was lost when they moved too fast. These nuclei were tracked again later in the sequence, but some links were not recovered (resulting into two splitted traces for one single nucleus motion). The tracking procedure is done by the reproduction behavior of the agent. The nuclei must overlap in successive frames to ensure the success of tracking.

The motions are different, and show particular patterns. In Figures 13a and b corresponding to natural motion, the cell's total displacement during the whole sequence is small. The cells have a random motion and seem to move inside a small area of the culture support. It is worth noting that the apparent displacement lengths are similar independently of the time interval between the frames (3 vs 30 min). Therefore, this displacement measured during 30 min in sequence NM3 (frames 1 and 11 only) is much smaller than the accumulation of the 10 displacements measured every three min (sequence NM3 with all frames from 1 to 11).

Beside this motion pattern, the two other sequences show a different displacement pattern of the cells. The kind of motion observed in Figures 13c and d differs during the evolution of the process. In the first phase (WC1), the cells close to the injury have a global motion oriented toward it. This

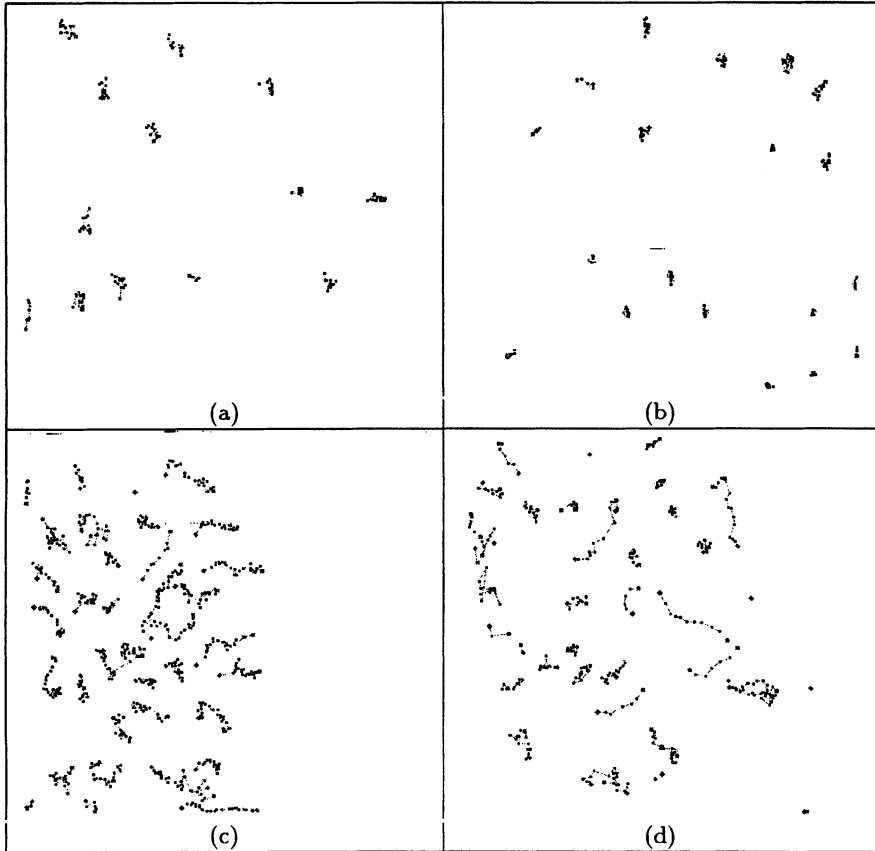


Figure 13. Cell tracking by their nuclei. (a) Natural motion (sequence NM30). (b) Natural motion (sequence NM3) (c) First phase of the wound process (sequence WC1). (d) Second phase of the wound process (sequence WC2).

suggests two different behaviors, depending on the position of the cells relative to the injury. In the second phase (WC2), the cells are spaced and some cells still have a directional motion. In the two phases, some cells do not follow the main observed motion but have a random displacement. It should be noted that random motion can have a directional component and that directional motion can have a random component. During the wound closure, a sequence of shorter time interval (3 min) would be of interest to determine more accurately the nature of the motion at higher frequency.

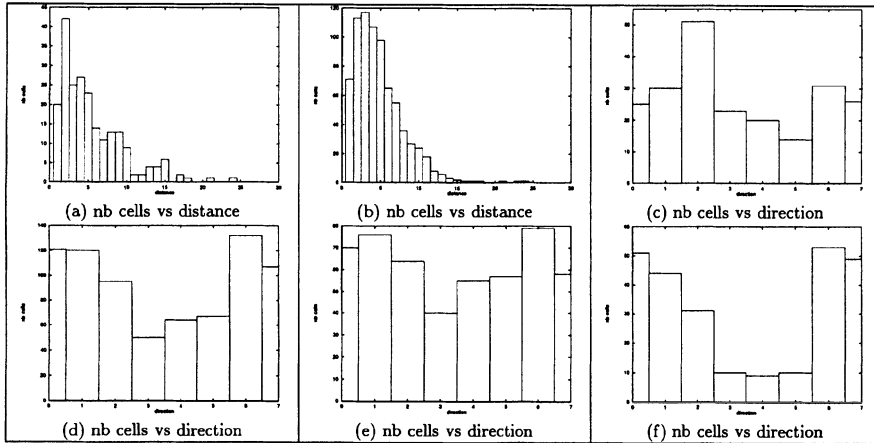


Figure 14. Histograms of distances and direction. (a) Histogram of distance covered for the natural motion (sequence NM30). (b) Histogram of distance covered for the first phase of the wound process (sequence WC1). (c) Histogram of motion direction for a natural motion (sequence NM30). (d) Histogram of distance covered for the first phase of the wound process (sequence WC1). (e) Histogram of motion direction for the first phase of the wound process (cells far from the injury) (sequence WC1). (f) Histogram of motion direction for the first phase of the wound process (cells close to the injury) (sequence WC1). All data (distance and direction) are computed between two successive frames. Distances are in pixels and directions are coded upon the Freeman code.

3.4 Analysis of the migration process

3.4.1 Distance

Migration rates of fibroblasts were determined by calculating the nuclear displacement for individual cells. From Figures 14a and b, one can see that the distances covered during natural motion or during the wound process are almost the same. After calibration, the mean speed of the cells during natural motion (sequence NM30) and wound closure (sequence WC1) are respectively $4.6 \mu\text{m}/\text{hour}$ and $4.3 \mu\text{m}/\text{hour}$ (Figure 14a and b). Those values can be compared to the value of $1.5 \mu\text{m}/\text{hour}$ obtained by Dunlevy and Couchman (1993), with rat embryo fibroblasts treated by heart-conditioned medium.

3.4.2 Direction

Directions in histograms were computed using the Freeman code. In this code, distances are rounded to 45° and labeled from 0 to 7, 0 being a left displacement, 2 an upward displacement, etc. . . . From the comparison of the direction histograms of natural motion (NM30) and wound closure (WC1) sequences (Figures 14c and d), a preferential direction can be observed for the second one.

Figures 14e and 14f show that nearly all the cells near the injury's side clearly move toward 0, 1, 2, 6 and 7 directions i.e. toward the denuded area of the cell monolayer. This tendency is not so evident for the cells inside the cell sheet. In the first case, cell migration can be the result of a directional *voluntary* cell displacement. It could also be explained by a random movement into a denuded area. However, from Figure 13c we can see that some cells near the injury's side still have a nearly straight displacement; cells being more widely spaced, random motion would be doubtful.

Here, only primary motion analysis methods have been presented. Further extension to this analysis is under study to show potential correlations between motion and some shape factors of the cells. Only nucleus components have been studied to date. The integration of all the cell's components will improve the characterization of cell motion.

4. Discussion

The present system would be of interest for studying the influence of various substances such as growth factors on the wound closure process. This model could also be modified in order to more accurately reflect the *in vivo* system. For instance, cells could be cultured on a fibronectin coated support to specify the importance of extracellular matrix components during cell migration. Cell migration could also be studied using chemo-attractant substances or oriented gels leading to a contact guidance of cells (Dickinson et al. 1994). It would thus be possible to analyse whether factors promoting autonomous migration of cells into a denuded area of a cell monolayer differ from those responsible for their chemotactic or biased migration (Kondo et al. 1993). These different models could provide a better knowledge of *in vivo* cell migration mechanisms involved in processes such as wound healing and tumor cell invasion.

The proposed agent model is similar to the one presented by Guessoum and Dojat (Guessoum and Dojat 1996) and applied to breath monitoring and economic simulation. Both models are multi-level and integrate various behaviors of different types, reactive (perception, reproduction) and cognitive (interaction). Two levels of competition are present in the system. Agents compete with themselves in the segmentation of an image. They can label, merge or negotiate zones. The behaviors interact asynchronously and concurrently under the control of a internal manager, which supervises the global time scheduling. Behaviors into one agent also work under a competition scheme. They are driven by external and internal events. These events modify their priorities and allow one of them to execute at a given time.

Our system integrates various levels of analysis, from the segmentation to cell tracking. The segmentation is specialized for the different compo-

nents of the images. It relies on sets of criteria depending on both static and motion-based information. We are attempting to make these criteria more adaptative to the local situation in an image. For exemple, an adaptative and evolutive thresholding is used for the nuclei, computed dynamically based on the nuclear mean gray level and a threshold factor. Another improvement would be to analyse the gray level profiles around the nuclei, according to a few predefined directions, and to adapt the threshold factor based on this profile. The reproduction strategy can be improved in the same way. In the current system, an agent launches a seed around itself at a prefixed distance. This launching strategy could be easily improved by an exploration phase of the environment, based on the previously mentionned nucleus surrounding profiles to find the best location for the new agent. This would imply an adaptative reproduction depending on the type of agent to be initiated. For the reproduction toward the next frame, the process could estimate the future position of the agent using the motion information computed so far.

The possibilities of motion analysis are reduced to tracking the nuclei through the sequence and computing some statistics on their motion. For further work, a cellular entity will be created to organize the agents working on the same cell. This higher abstraction level will give the opportunity to build a more complete analysis of cell behavior. One cell component motion can be decomposed into two motions: the global cell motion and the specific motion of the component. By studying these two motions separately, a more complete interpretation of the complete cell motion can be made.

In this paper, the wounding injury process was presented as an example of a typical application in which an intelligent system can be used.

References

- Baujard, O. & Garbay, C. (1993). KISS: a multiagent segmentation system. *Optical Engineering* **32**(6): 1235–1249.
- Beil, M., Irinopoulou, T., Vassy, J. & Rigault, J. P. (1995). Chromatin texture analysis in three-dimensional images from confocal scanning laser microscopy. *Analyt. Quant. Cytol. Histol.* **17**(5): 323–331.
- Bellet, F., Salotti, J. M. & Garbay, C. (1995). Une approche opportunistes et cooperative pour la vision de bas niveau. *Traitement du Signal* **12**(5): 479–494.
- Bernstein, J. J., Goldberg, W. J., & Law Jr., E. R. (1993). Migration of fresh human malignant astrocytoma cells into hydrated gel wafers in vitro. *J. Neurooncol.* **18**: 151–161.
- Bodor, N. S., Kiss-Buris, S. T. & Buris, L. (1991). Novel soft steroids: effects on cell growth in vitro and on wound healing in the mouse. *Steroids* **56**: 434–439.
- Boissier, O., Demazeau, Y., Masini, G. & Skaf, H. (1994). Une architecture multi-agents pour l'implémentation du bas niveau d'un système de compréhension de scènes. In *2ièmes journées francophones IAD et SMA*. Voiron (France).
- Brooks, R. A. (1993). Intelligence without representation. *Artificial Intelligence* **47**: 139–159.

- Cloppet-Oliva, F. & Stamon, G. (1996). Segmentation cooperative région/contour pour une analyse automatique d'images de cellules en culture. In *Proceedings of 10e congrès Reconnaissances des Formes et Intelligence Artificielle*, volume 2, 1063–1072.
- Cocqueuz, J. P. & Philipp, S. (1995). Analyse d'images: filtrage et segmentation. Masson.
- Dickinson, R. B., Guido, S. & Tranquillo, R. T. (1994). Biased cell migration of fibroblasts exhibiting contact guidance in oriented collagen gels. *Ann. Biomed. Eng.* **22**: 342–356.
- Dunlevy, J. R. & Couchman, J. R. (1993). Controlled induction of focal adhesion disassembly and migration in primary fibroblasts. *J. Cell Sci.* **105**: 489–500.
- Francisco, J., Pauwels, O., Simon, S., Gasperin, P., van Houtte, P., Pasteels, J. L. & Kiss, R. (1995). Computer-assisted morphonuclear characterization of radiotherapy-induced effects in MXT mouse mammary adenocarcinomas surviving earlier radiotherapy. *Int. J. Radiation Oncology Biol. Phys.* **32**(2): 409–419.
- Fu, K. S. & Mui, J. K. (1987). A survey on image segmentation. *Pattern Recognition* **13**(1): 3–16.
- Guessoum, Z. & Dojat, M. A real-time agent model in a asynchronous-object environment. In *Proceedings of MAAMAW*, 190–20.
- Gwydir, S. H., Buettner, H. M. & Dunn, S. M. (1994). Non-rigid motion analysis of the growth cone using continuity splines. *ITBM* **15**(3): 309–321.
- Jain, R. (1981). Extraction of motion information from peripheral processes. *IEEE Trans. on PAMI* **3**(5): 489–503.
- Karasek, M. A. (1989). Microvascular endothelial cell culture. *J. Invest. Dermatol.* **93**: 33S–38S.
- Kondo, H., Matsuda, R. & Yonezawa, Y. (1993). Autonomous migration of human fetal skin fibroblasts into a denuded are in a cell monolayer is mediated by basic fibroblast growth factor and collagen, *In vitro Cell Dev. Biol.* **29A**: 929–935.
- Leitner, F., Paillason, S., Ronot, X. & Demongeot, J. (1995). Dynamic functional and structural analysis of living cells: new tools for vital staining of nuclear DNA and for characterization of cell motion. *Acta Biotheoretica* (43): 299–317.
- Lester, B. R. & McCarthy, J. B.. Tumor cell adhesion to the extracellular matrix and signal transduction mechanisms implicated in tumor cell motility, invasion and metastasis. *Cancer Metast. Rev.* **11**: 31–44.
- Leung, M. K. & Yang, Y. H. (1995). First Sight: A human body outline labeling system. *IEEE Trans. on PAMI* **17**(4): 359–377.
- Leymarie, F. & Levine, M. D. (1993). Tracking deformable objects in the plane using an active contour model. *IEEE Trans. on PAMI* **15**(6): 617–634.
- Liedtke, C. E., Gahm, T., Kappei, F. & Aeikens, B. (1987). Segmentation of microscopic cell scenes. *Analyt. Quant. Cytol. Histol.* **9**: 197–211.
- Lockett, S. J. & Herman, B. (1994). Automatic detection of clustered, fluorescent-stained nuclei by digital image-based cytometry. *Cytometry* **17**: 1–12.
- Maes, P. (1989). How to do the right thing. *Connection Science Journal* **1**(3): 291–323.
- Mareel, M. M., van Roy, F. M. & de Baetselier, P. (1990). The invasive phenotypes. *Cancer Metast. Rev.* **9**: 45–60.
- Matthay, M. A., Thiery, J. P., Lafont, F., Stampfer, M. F. & Boyer, B. (1993). Transient effect of epidermal growth factor on the motility of an immortalized mammary epithelial cell line. *J. Cell Sci.* **106**: 869–878.
- Muller, T., Geimer, P. & Bade, E. G. (1993). The growth factor-induced migration of epithelial cells is associated with a transient inhibiton of DNA synthesis and a downregulation of PCNA RNA. *Eur. J. Cell. Biol.* **60**: 123.
- Naudet, S., Nicolas, L., Faye, C. & Viala, M. (1996). Suivi temporel 3D d'objets en présence d'occultations dans une séquence d'images monoculaires. In *Proceedings of 10e congrès Reconnaissances des Formes et Intelligence Artificielle*, volume 2, 849–858.
- Paillason, S., Robert-Nicoud, M. & Ronot, X. (1995). Optimizing DNA staining by Hoechst 33342 for assessment of chromatin Organisation in living cells. In *Proceedings of Optical and Imaging Techniques in Biomedecine, SPIE Proceedings Series*, volume 2329, 225–235.

- Palcic, B. (1994). Nuclear texture: it can be used as a surrogate endpoint biomarker. *J. Cell Biochem.* **19**: 40–46.
- Parvin, B. A., Peng, C., Johnston, W. & Maestre, F. M. (1995). Tracking of tubular molecules for scientific applications. *IEEE Trans. on PAMI* **17**(8): 800–804.
- Person, J. M., Lovett, D. H. & Raugi, G. J. (1988). Modulation of mesangial cell migration by extracellular matrix components. inhibition by heparinlike glycosaminoglycans. *Am. J. Pathol.* **133**: 609–614.
- Rappolee, D. A., Mark, D., Banda, M. G. & Werb, Z. (1988). Wound macrophages express TGF-alpha and other growth factors in vivo: analysis by mRNA phenotyping. *Science* **241**: 708–712.
- Scharffetter, K., Stolz W., Lankat-Burttgereit, B., Mauch, C., Kulozik, M. & Krieg, T. (1989). In situ hybridization: a useful tool for studies on collagen gene expression in cell culture as well as in normal and altered tissue. *Virchows Arch. B. Cell Pathol.* **56**: 299–306.
- Schuh, A. C., Keating, S. J., Montecarlo, F. S., Vogt, P. K. & Breitman, M. L. (1990). Obligatory wounding requirement for tumorigenesis in v-jun transgenic mice. *Nature* **346**: 756–760.
- Schweitzer, H. (1995). Occam algorithms for computing visual motion. *IEEE Trans. on PAMI* **17**(11): 1033–1042.
- Siegert, F., Weijer, C. J., Nomura, A. & Miike, H. (1994). A gradient method for the quantitative analysis of cell movement and tissue flow and its application to the analysis of multicellular Dictyostelium development. *J. Cell Sci.* (107): 97–104.
- Sims, J. R., Karp, S. & Ingber, D. (1992). Altering the cellular mechanical force balance results in integrated changes in cell, cytoskeletal and nuclear shape. *J. Cell Sci.* **103**: 1215–1222.
- Smoot, E. C., Kucan, J. O., Roth, A., Mody, N. & Debs, N. (1991). In vitro toxicity testing for antibacterials against human keratinocytes. *Plast. Reconstr. Surg.* **87**: 917–924.
- Steels, L. (1990). Cooperation between distributed agents through self-organization. In *DAI*, volume 1, 175–196. Elsevier Science.
- Stokes, C. L., Rupnick, M. A., Williams, S. K. & Lauffenburger, D. A. (1990). Chemotaxis of human microvessel endothelial cells in response to acidic fibroblast growth factor. *Lab. Invest.* **63**: 657–668.
- van Luyn, M. J., van Wachem, P. B., Nieuwenhuis, P. & Jonjman, M. F. (1992). Cytotoxicity testing of wound dressings using methylcellulose cell culture. *Biomaterials* **13**: 267–275.
- Watanabe, S., Hirose, M., Yasuda, T., Miyazaki, A. & Sato, N. (1994). Role of actin and calmodulin in migration and proliferation of rabbit gastric mucosal cells in culture. *J. Gastroenterol. Hepatol.* **9**: 325–333.
- Wong, J., Wang, N., Miller, J. W. & Schuman, J. S. (1994). Modulation of human fibroblast activity by selected angiogenesis inhibitors. *Exp. Eye Res.* **58**: 439–451.
- Wu, K., Gauthier, D. & Levine, M. D. (1995). Live cell image segmentation. *IEEE Trans. on Bio. Eng.* **42**(1): 1–12.

Color Computer Vision and Artificial Neural Networks for the Detection of Defects in Poultry Eggs*

V. C. PATEL, R. W. McCLENDON and J. W. GOODRUM

*Department of Biological and Agricultural Engineering and Artificial Intelligence Center,
University of Georgia, Athens, Georgia, 30602-4435, USA (E-mail: rwmc@bae.uga.edu)*

Abstract. A blood spot detection neural network was trained, tested, and evaluated entirely on eggs with blood spots and grade A eggs. The neural network could accurately distinguish between grade A eggs and blood spot eggs. However, when eggs with other defects were included in the sample, the accuracy of the neural network was reduced. The accuracy was also reduced when evaluating eggs from other poultry houses. To minimize these sensitivities, eggs with cracks and dirt stains were included in the training data as examples of eggs without blood spots. The training data also combined eggs from different sources. Similar inaccuracies were observed in neural networks for crack detection and dirt stain detection. New neural networks were developed for these defects using the method applied for the blood spot neural network development.

The neural network model for blood spot detection had an average accuracy of 92.8%. The neural network model for dirt stained eggs had an average accuracy of 85.0%. The average accuracy of the crack detection neural network was 87.8%. These accuracy levels were sufficient to produce graded samples that would exceed the USDA requirements.

Key words: color computer vision, neural networks, machine vision, egg grading, blood spots, dirt stains, cracks

Introduction

In modern egg processing plants, the inspection of eggs for defects (or grading) is a major bottleneck because it is largely done by human workers. Automated detection of cracked eggs is performed in a very limited number of plants, but currently no practical system for detecting blood spots and dirt stains exists. In order to obtain maximum throughput, processing speeds of over 85,000 eggs per hour are common. The demanding requirements placed on the human workers result in two types of grading errors. Overpull occurs when grade A eggs are graded as defective and underpull is when defective eggs are allowed to be included as grade A eggs. The egg producer

*This study was supported by State and Hatch funds allocated to the Georgia Agricultural Experiment Stations and grant funds from the Southeastern Poultry and Egg Association. The use of trademarks does not indicate endorsement of the product by the authors.

must minimize overpull and underpull to maximize profits and meet USDA requirements designed to maintain the quality of the product. An automated system capable of detecting eggs with blood spots, dirt stains, and cracks would be desirable since it could reduce the work load on human graders, increase the profitability of the egg producer, and improve the quality control process.

Blood spots are internal egg defects due to hemorrhaging in the ovaries during ovulation, salmonella infection, genetics, and seasonal factors [1]. North and Bell [2] attributed blood spots also to factors such as feed and the age of the hens. The albumen in fresh eggs is frequently cloudy, making the detection of blood spots more difficult [3]. North and Bell [2] estimated the average frequency of blood spots to be 0.9%. Eggs with small blood spots less than 0.32 cm (0.13 in.) in diameter (aggregate) must be classified as grade B [4]. Eggs with larger blood spots must be classified as "loss" and be discarded. Frequently such eggs are used by the animal feed industry.

Moisture and dirt accumulation on cage floors are a cause of dirt stained eggs [2]. Egg stains may also be attributed to bleeding during egg laying and fecal matter. Little research has been done on the incidence of dirt stained eggs. In modern egg processing facilities, eggs are washed prior to grading. However, some stains may remain. Stains may also occur after washing due to the presence of other severely cracked eggs on the processing line. USDA regulations require that eggs be classified as dirty if they have moderate stains, localized stains covering not more than 1/32 of the shell surface area, or scattered stains covering not more than 1/16 of the shell surface area [4]. Bourely et al. [5] estimated a dirt-stain frequency of 1%.

North and Bell [2] estimated that between 3% and 5% of eggs are cracked before processing. Factors such as genetics, age of the hen, amount of handling during processing, environmental temperature, diseases, and humidity influence the frequency of cracks [2]. Crack frequency can range as high as 10% in cases where the flock is aged or with collection equipment problems (Dr. Danis Cunningham, August 3, 1995. Personal Communication. Professor, Poultry Science, University of Georgia, Athens, GA).

According to the USDA egg grading manual [4], a sample of grade A eggs, after grading at the processing plant, must consist of at least 87% A quality or better eggs. Of the 13% that may be of a lower quality, 5% may be checks (cracks), 1% may be grade B due to air cells, blood spots less than 0.32 cm (0.13 in.) diameter (aggregate), or other yolk defects, and 0.5% may be leakers, dirties, or loss eggs in any combination. Leakers, dirties, or loss eggs may not constitute more than 0.3% individually.

Freeman [6] discussed machine vision systems for inspection. The discussion included sensors, illuminators, and processing systems. D'Agostino [7]

developed a custom machine vision system for the inspection of food. Applications of the system included determining the size and grade of citrus, and the inspection of processed meat for defects such as discoloration. Anand et al. [8] investigated the costs of a machine vision system station for grading produce. Lighting methods, cameras, and frame grabbers were discussed and their costs evaluated. The paper included a discussion on the issues that must be considered for a color-based machine vision system. Heinemann et al. [9] used machine vision to grade mushrooms based on color, shape, stem cut, and opening of the cap veil. The system was 80% accurate on average. Scanlon et al. [10] developed a computer vision system to quantify the color of potato chips. They used mean grey-scales of images to successfully detect differences in the color of potato chips.

Gittins and Overfield [11] studied alternative methods for grading eggs and developed an electronic system for measuring various characteristics of an egg such as weight, color, albumen quality, yolk color, and shell density. Elster and Goodrum [12] developed a program to analyze grey-scale images of stationery eggs for cracks. The egg was isolated from background noise and enhanced using image processing algorithms. A 96% success rate was achieved. However, the average time required to process one egg was 25.3 seconds. Goodrum and Elster [13] extended their work to detect cracks at any point on the surface of rotating eggs. The identification of cracks was dependent on the egg size and required software calibration constants.

Bullock et al. [14] provided a brief tutorial on artificial neural networks and discussed two applications – inspection of cookies for damage, and inspection of apples for bruises. The use of artificial neural networks in agriculture was discussed by Davidson and Lee [15]. Various application areas and potential uses such as planning, harvesting, sorting and inspection, image analysis, and the control of processing plants, were outlined. Timmermans and Hulzebosch [16] developed a color computer vision system for on-line inspection of flowers and ornamentals. The system used both statistical and neural networks for the classification of the plants. An on-line learning feature was also implemented. Alchanatis and Searcy [17] implemented a system for the inspection of carrots for shape and surface defects. The system used neural networks to classify carrots into two classes. Using a pipelined image processing system, grading speeds of 2 carrots per second were achieved with an accuracy of over 90%.

Patel et al. [18] used image acquisition routines from the work of Elster and Goodrum [12] to capture grey-scale images of cracked and grade A eggs. Histograms of the images were generated and used to train a neural network for the detection of cracked eggs. The model was 90% accurate and provided significant improvement in speed over the method of Elster and Goodrum

[12]. The work was extended to the detection of blood spots and dirt stains [19]. The neural network model for blood spot detection was 85.6% accurate. An accuracy of 80% was achieved on dirt stain detection.

Goal and Objectives

The overall goal of this research was to develop a coupled color computer vision and neural network system for detection of eggs with defects. The objectives of this research project were as follows:

- 1) to develop neural network models capable of differentiating eggs with a particular defect from eggs without that defect,
- 2) to develop robust neural network models with minimized sensitivity to eggs from different sources, and
- 3) to evaluate the computer vision and neural network system by comparing its accuracy to USDA requirements for egg processing plants and to the accuracy previously obtained with grey-scale images.

Materials and Methods

A color video camera and a color image acquisition board (frame grabber) were used to obtain color images. A Speed KingTM 25 W incandescent candling lamp was used to back-illuminate the egg. The lamp generated a light with an intensity of approximately 11000 lx. The image sensor was a SonyTM 3-chip CCD video camera (model DXC-930) with a horizontal resolution of 0.125 mm/pixel, and a vertical resolution of 0.110 mm/pixel. The camera was equipped with a CanonTM (YH17×7KTS) automatic iris lens with a focal length of 55 mm. A close-up lens (CanonTM model 82CL-UP800H) with a focal length of 800 mm was used to reduce the required lens-to-object distance. The distance between the lens system and the egg (object distance) was approximately 555 mm. The camera was connected to a Data TranslationTM DT2871 RGB/HSI color frame grabber that was used to capture the images. The board was capable of real-time capture and display of images at 30 frames per second in 16,581,375 colors. The color frame grabber had a horizontal resolution of 512 pixels and a vertical resolution of 480 pixels. The color frame grabber was installed in a 50 MHz 80486 IBMTM PC compatible computer. A SonyTM (model PVM-1340) color video monitor was used to observe the images of the eggs. Figure 1 shows a schematic of the imaging system.

The imaging system was used to obtain color images of defective eggs and grade A eggs. Histograms for the red, green, and blue colors were generated

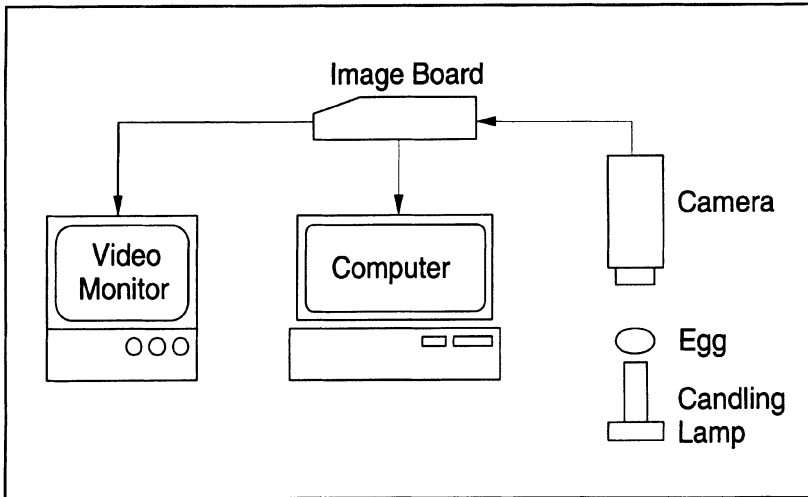


Figure 1. A schematic diagram of the imaging system.

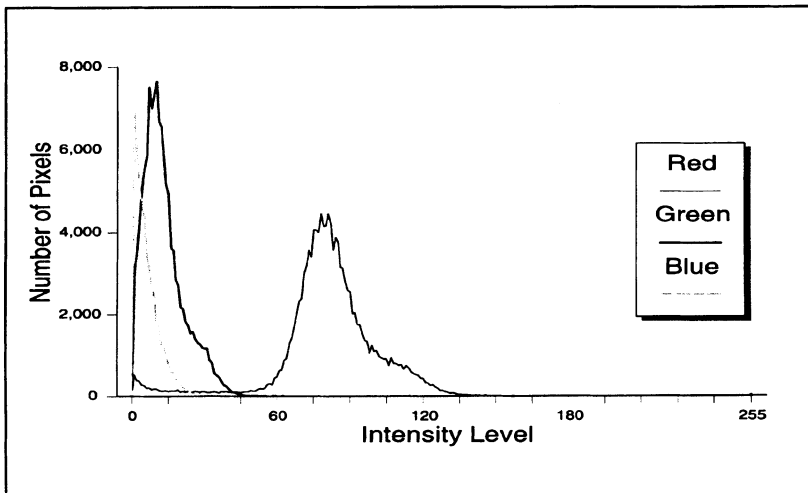


Figure 2. RGB histograms of a typical grade A egg.

from the images by counting the number of pixels at each intensity level. Since there were 256 intensity levels, this generated three histograms with 256 cells each. Figure 2 shows typical red, green, and blue histograms (with 256 cells) of a grade A egg. Although these typical histograms showed no apparent pixel counts in the region 180–255, other histograms had pixel

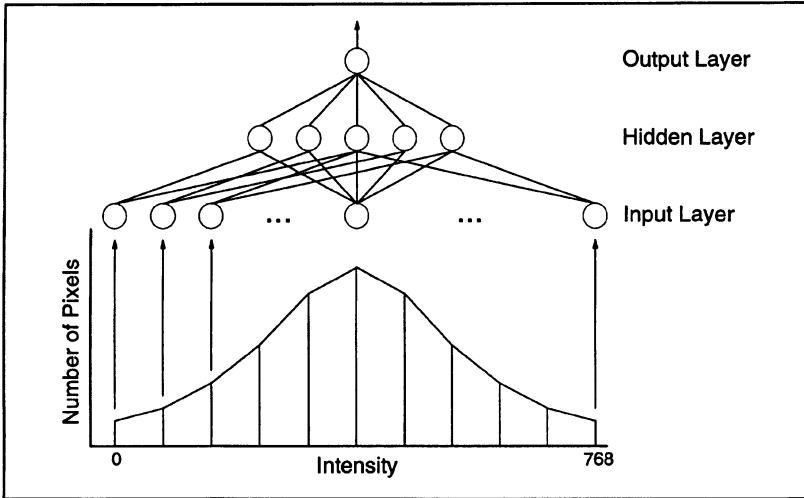


Figure 3. Histogram inputs to a neural network.

counts in this range. There were also no apparent patterns in the histograms of defective eggs which could be used to limit the range of the histogram cells. The three histograms were then joined to form a composite histogram with 768 cells and the number of pixels in the cells were used as inputs to a neural network (Figure 3). A commercial neural network simulator from Ward Systems Group, NeuroShellTM 2 [20], was used in the training and testing of the neural network models. NeuroShellTM 2 determines an optimal network by evaluating the predictive capability of the current neural network on an independent testing set. If the average error of the current neural network during training is less than the average error of the previous optimal neural network, the weights of the current neural network are saved as the new optimal neural network. The optimal network feature aids the user in determining when to stop training and thus develop a neural network with the maximum generalization ability. The current neural network is also saved at the end of the training session.

Professionally graded samples of 180 blood spot eggs and 180 USDA grade A eggs were obtained. The sample of blood spot eggs was exclusive to this defect. Color images of all eggs were obtained. Histograms of the red, green, and blue colors were generated from the egg images, concatenated into composite histograms, and transformed into neural network input patterns. A training set of 180 patterns was constructed by randomly selecting from the set of patterns of the blood spot eggs and the grade A eggs. From the remaining 180 patterns, another 90 patterns were randomly selected to comprise the

testing set. The rest of the patterns (90) formed the validating set. The training, testing, and validating sets were constrained to have equal numbers of blood spot egg patterns and grade A egg patterns. Training, testing, and validating data for dirt stained eggs were similarly generated.

Preferred values of the neural network parameters (learning rate and momentum) were determined. Learning rate and momentum parameter values of 0.1, 0.6, and 0.9 were considered. The initial neural network structure was 768 inputs, 56 hidden nodes, and 1 output. The number of hidden nodes was varied to determine a suitable number for generalization. The values considered were 8, 24, 40, 56, 72, and 104 hidden nodes. For these models, the neural network parameter values obtained previously were used. The number of inputs was reduced to 384 by combining two adjacent cells in each histogram, and the number of hidden nodes was varied again to determine an effective neural network structure. A neural network with fewer inputs and hidden nodes is desired because it would require less computer resources. Training was stopped when either the average error on the training set was less than a preset value, or 100,000 learning events had elapsed since an optimal network was last determined, or the total number of learning events exceeded 500,000. A similar procedure was used in training and evaluating the neural network models for dirt stain detection and crack detection.

Results and Discussion¹

The accuracy of the neural network models for blood spot detection, dirt stain detection, and crack detection was determined by applying their respective training, testing, and validating data sets to the neural network models after training was complete. Table 1 shows the accuracy of the neural networks as well as their structure in terms of the number of inputs, hidden nodes, and outputs. The blood spot detection neural network had an accuracy of 91.1% using 384 inputs and 24 hidden nodes. The highest accuracy achieved by the dirt stain detection neural network was 97.8% using 384 inputs and 40 hidden nodes. The most accurate crack detection neural network from a previous study [21] had an accuracy of 96.7%. That neural network had 384 inputs and 24 hidden nodes.

Analysis of accuracy

The USDA requires no more than 1% of grade A eggs be of B quality due to air cells, blood spots, or yolk defects. If we assume that there is an equal proportion of these defects then the final graded sample can have no more than 0.33% blood spotted eggs and no more than 0.33% dirt stained eggs. The

Table 1. Results of neural networks for detection of dirt stained and cracked eggs

Neural network	Egg defect	Network structure ¹	Learning events	Classification accuracy ²		
				Training set	Testing set	Validating set
1.1	Blood spots	384-24-1	14300	99.4 (0/1)	93.3 (3/3)	91.1 (3/5)
1.2	Dirt stains	384-40-1	11640	100 (0/0)	96.7 (1/2)	97.8 (2/0)
1.3	Cracks	384-24-1	28480	100 (0/0)	97.8 (1/1)	96.7 (2/1)

¹ Number of inputs-Number of hidden nodes-Number of outputs

² % Correct (no. overpull/no. underpull)

final graded sample should also have less than 5% cracked eggs [4]. Taking a sample of 10,000 eggs of which 0.9% have blood spots, 1% are dirt stained, and 5% are cracked, which are average defect frequencies, there would be 90 eggs with blood spots, 100 dirt stained eggs, and 491 cracked eggs. The neural network model for detection of blood spots in eggs (Model 1.1²) was 89.9% accurate and so would pull 80 of the 90 blood spotted eggs. The neural network model for dirt stain detection (Model 1.2) had an accuracy of 100% on dirt stains and would therefore pull all 100 dirt stained eggs.

Since the crack detection model (Model 1.3) had 97.8% accuracy on cracked eggs, it would correctly identify 481 of the 491 eggs with cracks in the sample. A sample of 10,000 eggs would consist of 9,319 grade A eggs (10,000-90-100-491). The blood spot detection neural network model was 93.3% accurate on grade A eggs and so would pull 6.7% of the grade A eggs in the sample (overpull). Similarly, the neural network models for dirt stained eggs and cracked eggs both would have an overpull of 4.4%. Therefore, 1,374 grade A eggs would be pulled as overpull. The percentage of eggs with blood spots in the final graded sample would be 0.126% which is within the USDA requirement of 0.33%. Since the neural network model for dirt stain detection was 100% accurate on dirt stained eggs, there would be no dirt stained eggs in the final graded sample. The percentage of cracked eggs in the sample would be 0.126%.

Interactions of neural network models

In an actual implementation, the blood spot detection neural network would be required to inspect all eggs (i.e. blood spotted, cracked, dirt stained, and grade A eggs) when checking for blood spots. To test the accuracy of the blood spot detection neural network on other defects, it was evaluated on the training, testing, and validating data used in developing the dirt stain detection (Model 1.2) and crack detection (Model 1.3) neural networks. The blood spot neural network (Model 1.1) was chosen for this study. Similarly, the dirt stain detection neural network (Model 1.2) was evaluated on the training,

Table 2. Accuracy of neural networks trained to distinguish between eggs with a specific defect and grade A eggs, evaluated on eggs with other defects

Neural network	Egg effect	Data set	Classification accuracy ¹	
			Grade A eggs	Defect ² eggs
2.1	Blood spot	Crack	82.2 (32)	71.7 (51)
		Dirt stain	84.4 (28)	25.0 (135)
2.2	Crack	Blood spot	97.2 (5)	22.8 (139)
		Dirt stain	98.9 (2)	82.8 (31)
2.3	Dirt stain	Blood spot	65.6 (62)	61.7 (69)
		Crack	96.7 (6)	95.6 (8)

¹ % Correct (no. incorrect)

² Defect type of data set

testing, and validating data used for the blood spot and crack detection neural networks. The crack detection neural network (Model 1.3) was evaluated on the training, testing, and validating data used for the blood spot and dirt stain detection neural networks. The results are shown in Table 2.

All the neural networks had a high accuracy on grade A eggs. However, the neural networks had varying degrees of accuracy when grading eggs with other defects. The results suggest that the neural networks may be differentiating between grade A and defective eggs but not differentiating between particular defects. For a neural network to be able to differentiate between defects, it must be presented with patterns which have the specific defect and patterns with other defects during the training phase. This was accomplished by including examples of eggs with other defects in the training, testing, and validating sets.

A neural network model for blood spot detection was developed with training, testing, and validating data consisting of eggs with blood spots as examples of defective eggs, and grade A eggs, cracked eggs, and dirt stained eggs as examples of eggs without blood spots. Neural network models for crack detection and dirt stain detection were also developed in this manner. The model development method discussed above was used to obtain the most accurate neural network models. Table 3 shows the accuracy of these neural networks on the training, testing, and validating data. The neural networks were evaluated on a new batch of grade A, blood spot, cracked, and dirt stained eggs with 200 samples of each. As shown in Table 4, the blood spot detection neural network could accurately distinguish between eggs with blood spots and eggs without blood spots. The high accuracy of the blood spot detection neural network supports the hypothesis of including eggs with other defects in the training data. The crack detection neural network had a

Table 3. Results of neural networks trained to distinguish between eggs with a specific defect eggs without that defect

Neural network	Egg defect	Learning events	Classification accuracy ¹		
			Training set	Testing set	Validating set
3.1	Blood spots	37000	99.4 (0/1)	84.4 (6/8)	90.0 (3/6)
3.2	Dirt stains	11560	93.9 (0/11)	96.7 (1/2)	95.6 (0/4)
3.3	Cracks	21860	98.3 (0/3)	90.0 (3/6)	88.9 (5/5)

¹ % Correct (no. overpull/no. underpull)

Table 4. Accuracy of neural networks trained to distinguish between eggs with a specific defect and eggs without that defect

Neural network	Egg defect	Classification accuracy ¹			
		Grade A eggs	Blood spot eggs	Cracked eggs	Dirt stained eggs
4.1	Blood spots	94.5 (11)	92.0 (16)	84.0 (32)	79.0 (42)
4.2	Cracks	82.0 (36)	98.5 (3)	63.0 (74)	63.5 (73)
4.3	Dirt stains	61.5 (77)	98.5 (3)	62.0 (76)	57.5 (85)

¹ % Correct (no. incorrect)

high accuracy on grade A eggs and blood spot eggs. However, its accuracy was reduced on cracked eggs and dirt stained eggs. The dirt stain detection neural network had a high accuracy on the blood spots eggs but not on grade A eggs and eggs with other defects. In this case, the neural networks had been trained on eggs from one poultry house and tested on eggs from another poultry house. To minimize the sensitivity of the neural networks to different defects and different eggs sources, the training, testing, and validating data were expanded to include eggs from different poultry houses.

Professionally graded samples of blood spot eggs, cracked eggs, dirt stained eggs, and USDA grade A eggs were obtained. All samples were constrained to have eggs with a single type of defect or of grade A quality. Color images of all eggs were obtained. Histograms of the red, green, and blue colors were generated from the egg images, concatenated into composite histograms, and transformed into neural network input patterns. A training set of 360 patterns was constructed by randomly selecting 180 patterns from the set of patterns of blood spot eggs, and 60 eggs from each of the cracked, dirt stained, and grade A eggs. A non-overlapping testing set of 180 patterns was constructed by combining 90 randomly selected patterns of blood spot eggs, with 90 randomly selected patterns of each of the cracked, dirt stained, and grade A eggs (30 from each category). A validating set of 180 patterns was similarly

Table 5. Results of training neural networks on combinations of a specific defect and other defects from two poultry houses

Neural network	Egg defect	Learning events	Classification accuracy ¹		
			Training set	Testing set	Validating set
5.1	Blood spots	90740	99.4 (0/2)	92.2 (5/9)	92.8 (4/9)
5.2	Cracks	34960	94.7 (10/9)	86.7 (12/12)	87.8 (8/14)
5.3	Dirt stains	60440	98.1 (7/0)	85.0 (17/10)	85.0 (14/13)

¹ % Correct (no. overpull/no. underpull)

Table 6. Accuracy of neural network models on specific types of defects

Neural network	Egg defect	Classification accuracy ¹			
		Grade A	Blood spots	Cracks	Dirt stains
6.1	Blood spot	93.3	90.0	93.3	100.0
6.2	Crack	93.3	90.0	84.4	90.0
6.3	Dirt stain	86.7	90.0	76.7	85.6

¹ % Correct

constructed. The training, testing, and validating data sets were constrained to include equal numbers of eggs from two different poultry houses. Training, testing, and validating data sets for developing crack detection and dirt stain detection neural networks were similarly generated.

A new set of neural networks for blood spot detection, crack detection, and dirt stain detection were trained. As before, experiments with the neural network learning parameters and structure were performed to obtain the most accurate neural network models as shown in Table 5. The average accuracy of the blood spot detection neural network on the validation set was 92.8%. The dirt stain detection and crack detection neural networks had average accuracies of 85.0% and 87.8%, respectively.

Table 6 shows the accuracy of the neural networks on grade A eggs and various defects in the validating set. The results indicate that the neural networks trained with all defects present and with eggs from various sources were more robust when grading eggs with other defects. The average accuracy of the neural networks was less than the accuracy obtained when the data were restricted to a single defect and grade A eggs. However, the models developed for these more realistic conditions were sufficiently accurate to generate graded samples that would exceed USDA requirements. Using a sample of 10,000 eggs, the percentage of blood spot eggs in the final graded sample was 0.113%. The percentage of dirt stained eggs and cracked eggs was 0.183% and 0.774%, respectively.

Using histograms of grey-scale images to train neural networks for detection of blood spots, dirt stains, and cracks resulted in accuracies of 85.6%, 80.0%, and 90.0%, respectively [18, 19]. The neural networks trained on histograms of color images had accuracies of 92.8%, 85.0%, and 87.8% for blood spots, dirt stains, and cracks, respectively. Although the color crack detection neural network was slightly less accurate than the grey-scale crack detection neural network, the color crack detection neural network was more robust in terms of inspecting eggs with other defects. Overall, the use of color computer vision improved the accuracy of the neural networks.

Conclusions

Neural networks trained entirely on eggs of one type of defect and grade A eggs could produce graded samples that would exceed USDA requirements. However, the neural networks were less accurate for different types of egg defects and also to eggs from different sources. To minimize these sensitivities, the training, testing, and validating data were modified to include examples of eggs with other defects and eggs from different poultry houses. The resulting neural networks were more robust and able to differentiate between the different types of defects.

The neural network model for blood spot detection had an average accuracy of 92.8% (90.0% on blood spot eggs and 95.6% on eggs without blood spots). The neural network model for dirt stained eggs had an average accuracy of 85.0% (85.6% on eggs with dirt stains and 84.4% on eggs without dirt stains). The average accuracy of the crack detection neural network was 87.8% (84.4% on eggs with cracks and 91.1% on eggs without cracks). These accuracy levels were sufficient to produce graded samples that would exceed the USDA requirements. The use of color computer vision improved the accuracy of the neural networks.

Acknowledgments

The authors acknowledge the assistance of Danis L. Cunningham, Professor, Poultry Science, University of Georgia, Bruce A. Webster, Assistant Professor, Poultry Science, University of Georgia, and Jerry L. Butler, Poultry Plant Supervisor, Poultry Science Research College, University of Georgia, who provided graded eggs and advice regarding the research project.

Notes

¹ Unless specified, all tables show the average accuracy of the optimal networks on the training, testing, and validating data as determined by the NeuroShell™ Optimal Network feature.

² Model numbers are based on the table number in which they appear and the position within the table, therefore Model 1.1 refers to the first entry in Table 1.

References

1. Wells, R. G. and Belyavin, C. G. (1987). *Egg Quality: Current Problems and Recent Advances*. Poultry Science Symposium Series Number Twenty. Butterworths: London, England.
2. North, M. O. and Bell, D. D. (1990). *Commercial Chicken Production Manual*, Fourth Edition. Van Nostrand Reinhold: New York, NY.
3. Stadelman, W. J. (1986). Quality Identification of Shell Eggs. In Stadelman, W. J. and Cotterill, O. J. (eds.) *Egg Science and Technology*. AVI Publishing Company, Inc.: Westport, CT.
4. United States Department of Agriculture (1990). *Egg-Grading Manual*. Agricultural Handbook Number 75, Agricultural Marketing Service, USDA.
5. Bourelly, A. J., Hsia, T. C. and Upadhyaya, S. K. (1986). Investigation of a Robotic Egg Candling System. In *Proceedings of the Agri-Mation2 Conference and Exposition*, 53–59. Chicago, Illinois.
6. Freeman, H. (1989). Machine Vision for Inspection. In Pieroni, G. G. (ed.) *Courses and Lectures, No. 307: Issues on Machine Vision*. International Center for Mechanical Sciences, Springer-Verlag: Wien, Italy.
7. D'Agostino, S. A. (1991). A Generic Machine Vision System for Food Inspection. In *Proceedings of the 1991 Symposium on Automated Agriculture for the 21st Century*, 3–7. Chicago, Illinois.
8. Anand, K. S., Morrow, C. T., Heinemann, P. H. and He, B. (1994). Development of a Low Cost Machine Vision Inspection Station for Grading Produce. In *1994 International Winter Meeting of the ASAE*. Atlanta, Georgia, Paper No. 943607.
9. Heinemann, P. H., Hughes, R., Morrow, C. T., Sommer III, H. J., Beelman, R. B. and Wuest, P. J. (1994). Grading of Mushrooms Using a Machine Vision System. *Transactions of the ASAE* 37(5): 1671–1677.
10. Scanlon, M. G., Roller, R., Mazza, G. and Pritchard, M. K. (1994). Computerized Video Image Analysis to Quantify Color of Potato Chips. *American Potato Journal* 71(11): 717–733.
11. Gittins, J. and Overfield, N. D. (1988). Computerization of Egg Quality Assessment. *World's Poultry-Science Journal* 44(3): 219–220.
12. Elster, R. T. and Goodrum, J. W. (1991). Detection of Cracks in Eggs Using Machine Vision. *Transactions of the ASAE* 30(1): 307–312.
13. Goodrum, J. W. and Elster, R. T. (1992). Machine Vision for Crack Detection in Rotating Eggs. *Transactions of the ASAE* 35(4): 1323–1328.
14. Bullock, D., Whittaker, D., Brown, J. and Cook, D. (1992). Neural Networks for Your Toolbox. *Agricultural Engineering* 73: 10–12, 31.
15. Davidson, C. S. and Lee, R. H. (1991). Artificial Neural Networks for Automated Agriculture. In *Proceedings of the 1991 Symposium on Automated Agriculture for the 21st Century*, 106–115. Chicago, Illinois.
16. Timmermans, A. J. M. and Hulzebosch, A. A. (1994). Optical Measurement System for On-Line Sorting of Ornamentals Using Neural Networks. In *Proceedings of the International Conference on Agricultural Engineering, AgEng '94*. Milano, Italy, Report No. 94-G-036.

17. Alchanatis, V. and Searcy, S. W. (1995). High Speed Inspection of Carrots with a Pipelined Image Processing System. In *1995 ASAE Annual International Meeting*. Chicago, Illinois, Paper No. 953170.
18. Patel, V. C., McClendon, R. W. and Goodrum, J. W. (1994). Crack Detection in Eggs Using Computer Vision and Neural Networks. *AI Applications* 8(2): 21–31.
19. Patel, V. C., McClendon, R. W. and Goodrum, J. W. (1996). Detection of Blood Spots and Dirt Stains in Eggs Using Computer Vision and Neural Networks. *Applied Engineering in Agriculture* 12(2): 253–258.
20. Ward Systems Group, Inc. (1994). *NeuroShell 2*. Ward Systems Group, Inc.: 245 West Patrick Street, Frederick, Maryland.
21. Patel, V. C., McClendon, R. W. and Goodrum, J. W. (1996). Detection of Cracks in Eggs Using Color Computer Vision and Artificial Neural Networks. *AI Applications* 10(3): 19–28.

Automatic Plankton Image Recognition

XIAOOU TANG¹, W. KENNETH STEWART¹, LUC VINCENT²,
HE HUANG¹, MARTY MARRA³, SCOTT M. GALLAGER¹ and
CABELL S. DAVIS¹

¹*Woods Hole Oceanographic Institution, Challenger Drive, MS #7, Woods Hole, MA 02543-1108, USA (Voice: 508-289-3226; Fax: 508-457-2191; Email: xtang@whoi.edu);*

²*Xerox Imaging Systems, 9 Centennial Drive, Peabody, MA 01960, USA;*

³*Vexcel Corporation, 2477 55th Street, Boulder, CO 80301, USA*

Abstract. Plankton form the base of the food chain in the ocean and are fundamental to marine ecosystem dynamics. The rapid mapping of plankton abundance together with taxonomic and size composition is very important for ocean environmental research, but difficult or impossible to accomplish using traditional techniques. In this paper, we present a new pattern recognition system to classify large numbers of plankton images detected in real time by the Video Plankton Recorder (VPR), a towed underwater video microscope system. The difficulty of such classification is compounded because: 1) underwater images are typically very noisy, 2) many plankton objects are in partial occlusion, 3) the objects are deformable and 4) images are projection variant, i.e., the images are video records of three-dimensional objects in arbitrary positions and orientations. Our approach combines traditional invariant moment features and Fourier boundary descriptors with gray-scale morphological granulometries to form a feature vector capturing both shape and texture information of plankton images. With an improved learning vector quantization network classifier, we achieve 95% classification accuracy on six plankton taxa taken from nearly 2,000 images. This result is comparable with what a trained biologist can achieve by using conventional manual techniques, making possible for the first time a fully automated, at sea-approach to real-time mapping of plankton populations.

1. Introduction

Plankton form the base of the food chain in the ocean and are a fundamental component of marine ecosystem dynamics. Understanding the ecological and physical processes controlling population dynamics of plankton over a wide range of scales, from centimeters to hundreds of kilometers, is essential for understanding how climate change and human activities affect marine ecosystems. Such studies require large-scale, high-resolution mapping of plankton abundance and taxonomic and size composition. High-resolution temporal sampling is needed to measure tidal, diel, and seasonal variability of population abundance and composition. Until recently, however, it has been difficult or impossible to conduct such extensive sampling because plankton abundance is highly variable in time and space and cannot be quantified with sufficient resolution using conventional sampling methods.

Traditionally, plankton surveys are conducted with such equipment as towed nets, pumps, and Niskin bottles. Because of the laborious deployment process and limited sample storage space on ship, the spatial sampling rate is extremely low. The painstaking and error-prone post-processing – manual counting of samples through a microscope and data entry – may take months or years, which effectively prohibits large-scale, high-resolution, three-dimensional surveys over periods of time. However, accurate estimates of production and growth can be made only if the interactions of organisms with one another and with the local environment are estimated from samples drawn at appropriate intervals of space and time (Owen 1989).

To help overcome the limitations of traditional plankton sampling instruments, a new Video Plankton Recorder (VPR) has been developed (Davis et al. 1992; Davis et al. 1992). As the VPR is towed through the water, it continuously captures magnified plankton images, providing a spatial resolution of plankton distribution on scales from microns to over 100 km. The amount of image data collected over even short periods of time can be overwhelming, necessitating an automated approach to plankton recognition. This approach would not only save a great deal of man power, but also make the real-time sorting of plankton possible. Real-time abundance and distribution data on zooplankton and accompanying environmental variables are needed to guide researchers during field studies on population and community processes, just as physical oceanographers have for decades used real-time measurements of temperature and conductivity to adjust their survey strategy according to observed phenomena (Paffenhofer 1991).

Now that high-quality images of individual plankton can be obtained with the VPR, our approach to the full automation of at-sea analysis of plankton size and taxonomic composition focuses on the development of an image analysis and pattern recognition system for real-time processing of the large volume of image data being acquired. Our development approach includes three parts (Davis et al. 1992; Davis et al. 1996): 1) a hardware/software system for preprocessing of the images (including real-time image capture, object detection, and in-focus analysis) and digital storage of detected object images; 2) pattern recognition algorithms for automated identification and classification of planktonic taxa; 3) incorporation of the pattern recognition algorithms into a high-performance image analysis system to achieve a real-time processing capability. Development of a preprocessing and acquisition system as described in Step 1 has been completed and used to detect and save subimages of planktonic taxa in real-time while at sea (Davis et al. 1996).

In this paper, we mainly address Step 2 and demonstrate an automated approach to plankton image classification. Our experimental data sets differ from those used for most previous pattern recognition researches in four

aspects: 1) the underwater images are much noisier, 2) many objects are in partial occlusion, 3) the objects are deformable, and 4) images are projection variant, i.e., the images are video records of three-dimensional objects in arbitrary positions and orientations. Figure 1, which shows example sub-images extracted from the larger video fields, illustrates the diversity of images within individual taxa.

By combining granulometric features with such traditional two-dimensional shape features as moment invariants and Fourier boundary descriptors, we extract a more complete description of the plankton patterns. Then, using an improved Learning Vector Quantization (LVQ) neural network classifier, we classify the plankton images into several taxonomic categories. The algorithms are tested on six classes taken from nearly 2,000 plankton images. The resulting classification accuracy is comparable with what a trained biologist can achieve using traditional manual techniques.

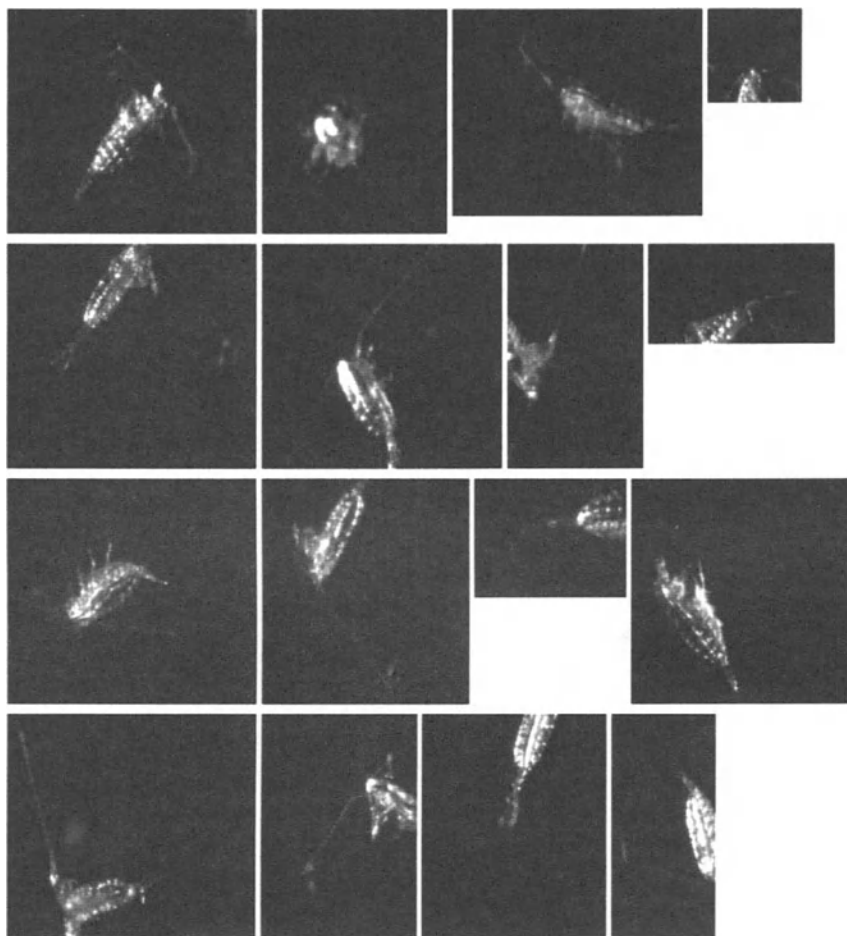
The paper is organized as follows. In Section 2, the three feature extraction methods – moment invariants, Fourier boundary descriptors, and granulometric features – are described, along with a feature selection algorithm. We then introduce an improved LVQ classifier. Section 3 describes real-time data acquisition and image processing. In Section 4, experimental results from the classification of the six plankton taxa are reported. We summarize our conclusions and point to future work in Section 5.

2. Methodology

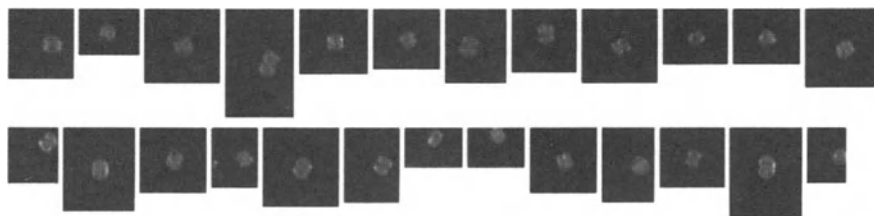
2.1. *Feature extraction*

Developing algorithms for classification of two-dimensional shapes insensitive to position, size, and orientation is an important problem in pattern recognition. Application of these algorithms range from industrial inspection and scene analysis to optical character recognition. The most widely used shape features are moment invariants and Fourier boundary descriptors. Classification of three-dimensional projection-variant objects is even more difficult.

In this paper, we introduce gray-scale granulometric features as a powerful pattern descriptor, which captures both shape and texture signatures, as a step toward addressing the three-dimensional problem. We also combine the three types of feature vectors to form a more complete description of the plankton patterns. We briefly review the three feature types then describe an effective feature selection method.

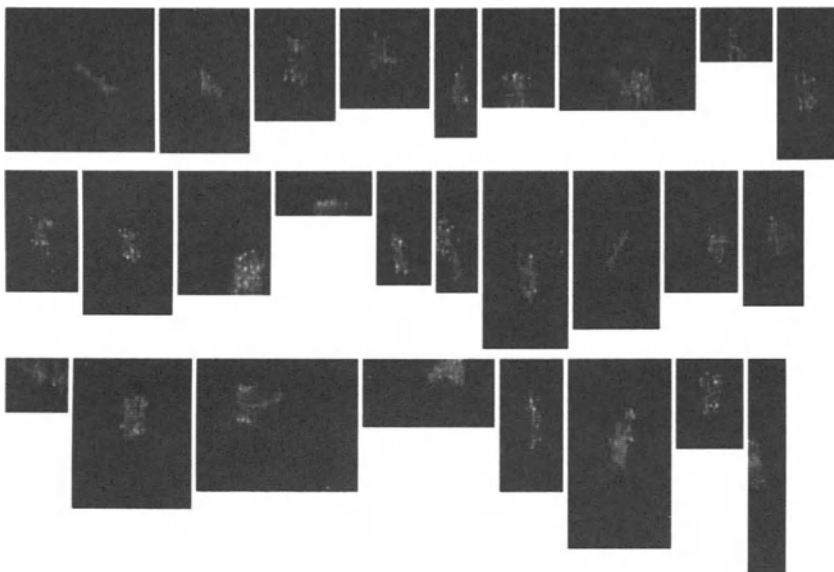


(a) CALANUS



(b) DIAT-CENTR

Figure 1(a-f). Sample images for each of the six types of plankton.

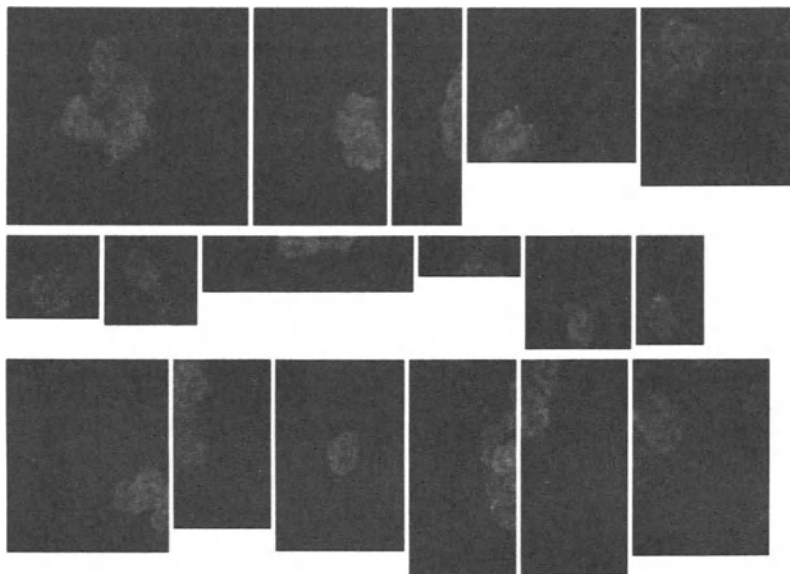


(c) DIAT-CHAET

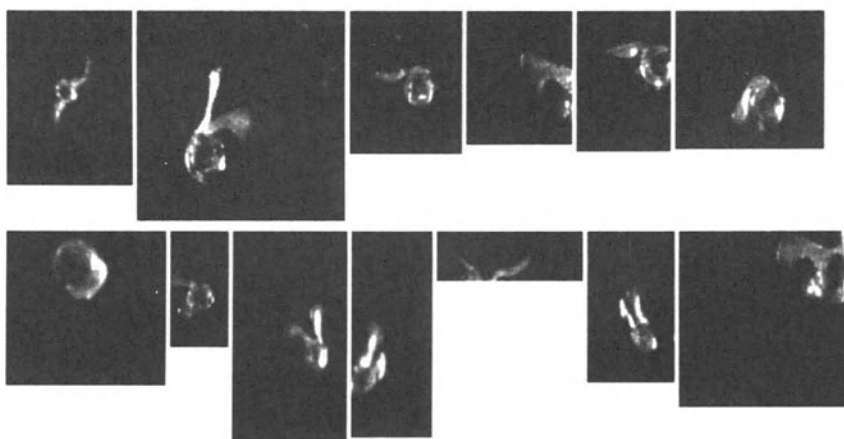


(d) DIATOM

Figure 1(a-f). Continued.



(e) DIATOMCOLO



(f) PTEROPOD

Figure 1(a-f). Continued.

2.1.1. *Moment invariants*

The concept of moments as invariant image features was first introduced by Hu (1962), and later revised by Reiss (1991). Moments and functions of moments have been used as pattern features in many applications. Some examples and comparisons of different features are found in Gonzalez and Wintz (1987), Reeves et al. (1988) and Teh and Chin (1991). In our experiments, we use the seven invariant moments described by Hu (1962), and Gonzalez (1987).

A $(p + q)$ th order moment of a continuous image function $f(x, y)$ is defined as

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad p, q = 0, 1, 2, \dots \quad (1)$$

For digital images the integrals are replaced by summations and m_{pq} becomes

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y). \quad (2)$$

The central moments of a digital image can be expressed as

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y), \quad (3)$$

where

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}}.$$

The normalized central moments are derived from these central moments as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma}, \quad \gamma = \frac{p+q}{2} + 1, \quad p+q = 2, 3, \dots \quad (4)$$

Based on methods of algebraic invariants, Hu (1962) derived seven invariant moments $\phi_i, i = 1 \dots 7$, using nonlinear combinations of the second and third normalized central moments. These invariant moments possess the desirable properties of being translation, rotation, and scale invariant.

2.1.2. *Fourier descriptor*

The first in depth study of the Fourier descriptor was given by Zahn and Roskies (1972) and later refined by Persoon and Fu (1977). Recently, more research effort has been devoted to shape classification based on Fourier descriptors (Reeves et al. 1988; Kauppinen et al. 1995; Reti and Czinege 1989). The most often used boundary models include the curvature function, centroidal radius, and complex contour coordinates. Kauppinen et al. (1995)

give a detailed experimental comparison of different models. We use the radius Fourier Descriptor (FD) and complex contour Fourier descriptor, which were shown to be the best among FDs tested in Kauppinen et al. (1995).

Consider a closed boundary defined by a closed sequence of successive boundary pixel coordinates (x_i, y_i) . The centroidal radius function expresses the distance of boundary points from the centroid (x_c, y_c) of the object,

$$r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}. \quad (5)$$

A complex contour coordinate function is simply the coordinates of the boundary pixels in an object centered coordinate system represented as complex numbers,

$$z_i = (x_i - x_c) + j(y_i - y_c). \quad (6)$$

Since both functions are computed around the centroid of the object, they are automatically translation invariant. To achieve rotation and scale invariance, a Fourier transformation of the boundary signature is generally used. For digital images we use the discrete Fourier transform (DFT). The shift invariant DFT magnitude gives a rotation invariant feature vector. Scale invariance is accomplished by normalizing all DFT magnitudes by the DFT magnitude of the zero frequency or the fundamental frequency component.

The feature vector for a radius Fourier Descriptor is

$$FD_r = \left[\frac{|F_1|}{|F_0|} \cdots \frac{|F_{N/2}|}{|F_0|} \right], \quad (7)$$

where N is the boundary function length and F_i denotes the i th component of the Fourier spectrum. Notice that only half of the spectrum need to be used because of the symmetric property of the Fourier transform of real functions.

The contour Fourier method transforms the complex coordinate function in Equation (6) directly. The feature vector is

$$FD_c = \left[\frac{|F_{-(N/2-1)}|}{|F_1|} \cdots \frac{|F_{-1}|}{|F_1|} \frac{|F_2|}{|F_1|} \cdots \frac{|F_{N/2}|}{|F_1|} \right]. \quad (8)$$

In this case, both positive and negative halves of the complex spectrum are retained. Because the complex function is centered around the coordinate origin, F_0 has zero value and F_1 is used for the normalization.

2.1.3. Granulometric features

The above traditional features are mostly developed in a well controlled pattern environment. Test images are usually shifted, scaled, and rotated

versions of a small set of perfect images of such simple objects as hammers, scissors, airplane silhouettes, and letters. However, most boundary features do not perform well when a small amount of noise is added to the original images (Kauppinen et al. 1995). As mentioned earlier, our experimental data sets are not only much noisier, but the plankton are non-rigid, projection-variant, and often in partial occlusion.

To overcome or at least partially alleviate these difficulties, we turned to features based on mathematical morphology (Matheron 1975; Serra 1982) known as granulometries. Granulometries were introduced in the sixties by Matheron as tools to extract size distributions from binary images (Matheron 1975). The approach is to perform a series of morphological openings of increasing kernel size and to map each kernel size to the number of image pixels being removed during the opening operation at this kernel size. The resulting curve, often called the pattern spectrum (Maragos 1989), maps each size to a measure of the image parts with this size. Typically, the peak of this curve provides the dominant object size in the image. For examples of the application of granulometries, refer to (Vincent 1994a, 1994b).

Beyond pure size information, granulometries actually provide a “pattern signature” of the image to which they are applied and can be used successfully as elements of a feature vector for shape classification problems (Schmitt and Mattioli 1991). Furthermore, the concept of granulometries can easily be extended to gray-scale images. In this context, granulometric curves capture information on object texture as well as shape. Additionally, various types of gray-scale granulometric curves can be computed, depending on the underlying family on openings or closing used. For example, curves based on openings with line segments capture information on bright linear image features, whereas curves based on closings with disk-shaped elements capture information on dark, “blobby” image parts.

To capture information both on linearly elongated and “blobby” image parts, whether dark or bright, we use four types of gray-scale granulometries. These curves are normalized by the total image volume removable by opening or closing, so that they are invariant with respect to illumination and scale changes. For example, an image of the copepod *Oithona* gives rise to a corresponding pattern spectrum characterizing bright, blobby image parts in Figure 2. The peak of this curve characterizes the size and contrast of the body of the organism. A similar curve results for an organism known as a pteropod (a planktonic snail) in Figure 3. The corresponding granulometric curve however, is distinctly different from the previous one, reflecting the more complex body configuration of the pteropod. Several novel gray-scale granulometry algorithms are used to compute the granulometric curves. These

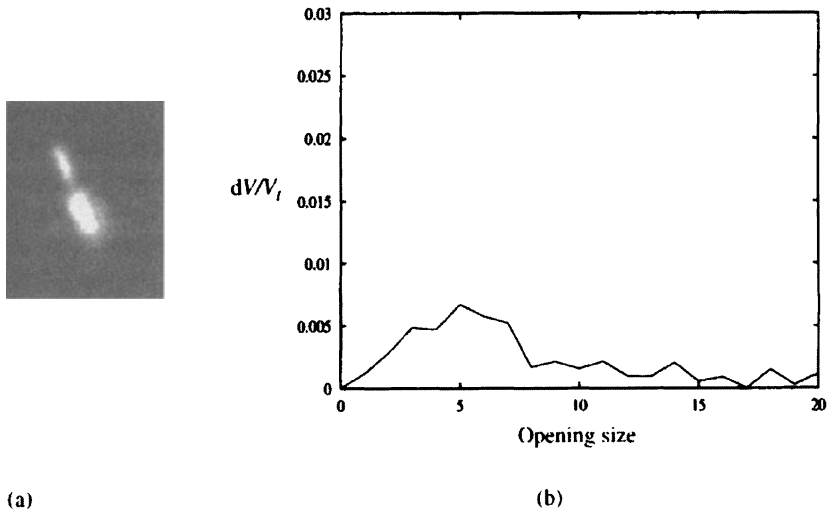


Figure 2. Copepod *Oithona* image (a) and corresponding granulometric curve (b), where dV is the decrease of image “volume”, i.e. the sum of pixel values, from one opening to the next, and V_t is the total volume removable by opening.

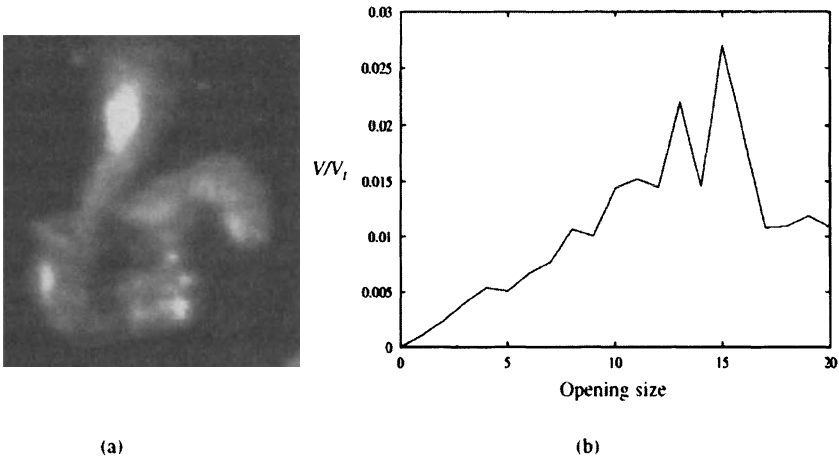


Figure 3. Pteropod image (a) and corresponding granulometric curve (b).

algorithms are orders of magnitude faster than traditional techniques (Vincent 1994b), making possible for real-time application.

2.1.4. Feature selection

Very large feature vectors usually contain much redundant information. Processing requires significant time or computational power, and classifi-

cation results often are poor. For these reasons, decorrelation and feature selection steps are needed. The widely used Karhunen-Loeve transform (KLT) is an ideal feature reduction and selection procedure for our system. Its decorrelation ability serves to decorrelate neighborhood features, and its energy packing property serves to compact useful information into a few dominant features. For a large feature vector, however, the computation of the eigenvectors of the covariance matrix can be very expensive. The dominant eigenvector estimation method, described in Tang (1996) is used to overcome this problem.

As optimal representation features, KLT selected features may not be the best for classification. Additional feature class separability measures are needed to select KLT decorrelated features. We use the Bhattacharyya distance measure in this study. An analytical form of this measure is (Fukunaga 1972)

$$\beta = \frac{1}{8}(\mu_1 - \mu_2)^T \left(\frac{\sigma_1 + \sigma_2}{2} \right)^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \ln \frac{\left| \frac{1}{2}(\sigma_1 + \sigma_2) \right|}{|\sigma_1|^{1/2} |\sigma_2|^{1/2}}, \quad (9)$$

where σ_i and μ_i are the class variance and mean. From (9), we see that β is proportional to both the distance of class means and the difference of class variances. The feature selection criterion is to retain only those decorrelated features with large β value.

2.2. Classification algorithm

We use the learning vector quantization classifier (Kohonen 1987, 1990) for feature classification because it makes weaker assumptions about the shapes of underlying feature vector distributions than traditional statistical classifiers. For this reason, the LVQ classifier can be more robust when distributions are generated by nonlinear processes and are strongly non-Gaussian, similar to the case for our data.

A vector quantization process should optimally allocate M codebook reference vectors, $\omega_i \in R^n$, to the space of n -dimensional feature vectors, $x \in R^n$, so the local point density of the ω_i can be used to approximate the probability density function $p(x)$ (Kohonen 1987). Consequently, the feature vector space is quantized into many subspaces around ω_i , the density of which is high in those areas where feature vectors are more likely to appear and coarse in those areas where feature vectors are scarce.

To use such a vector quantization process in a supervised pattern classification application, Kohonen (1987, 1990) developed the Learning Vector Quantization classifier. First, the codebook reference vectors ω_i s are initialized by either M random feature vector samples or the mean value of the feature

vectors. They are then assigned to a fixed number of known application-specific classes. The relative number of codebook vectors assigned to each class must comply with the a priori probabilities of the classes. A training algorithm is then used to optimize the codebook vectors. Let the input training vector x belong to class C_t , and its closest codebook vector ω_r be labeled as class C_s . The codebook vector ω_i is updated by the learning rules (Kohonen 1987),

$$\begin{aligned} \delta\omega_r &= \alpha(x - \omega_r) & \text{if } C_s = C_t \\ \delta\omega_r &= -\alpha(x - \omega_r) & \text{if } C_s \neq C_t, \\ \delta\omega_i &= 0 & \text{for } i \neq r \end{aligned} \quad (10)$$

where α is the learning rate. Only the closest of the vectors ω_i is updated, with the direction of the correction depending on the correctness of the classification. Effectively, these codebook vectors are pulled away from zones where miscalculations occur, i.e., away from the classification boundary region. After training, the nearest neighbor rule is used to classify the input test vector according to the class label of its nearest codebook vector.

A neural network architecture to implement the LVQ is shown in Figure 4. The network consists of two layers, a competitive layer and a linear output layer. The weights connecting all input neurons with the competitive layer neuron i form the codebook vector ω_i . The net input for each competitive layer neuron is the Euclidean distance between the input vector x and the weight vector ω_i . The output of each neuron is 0 except for the “winner” neuron, whose weight vector has the smallest distance to the input vector and whose output is 1.

The second layer transforms the competitive layer’s neuron class into the final output class. As discussed above, the competitive layer neurons, i.e., the codebook vectors, are assigned to the output classes according to a priori probabilities of the classes. For the example shown in Figure 4, the first three neurons are assigned to Class 1, the next two to Class 2, and the final two to Class 3. Only the non-zero weight connections between the competitive layer and the linear layer are shown in the figure. If any neuron in a particular class wins, the corresponding neuron class in the linear output layer will have an output of 1.

A drawback of the LVQ algorithm is the time consuming training process. To improve it, we use a statistical initial condition and a parallel training strategy introduced by Tang (1996). First, we initialize the neuron weight vectors in the competitive layer using the means and variances of the training vectors in each class. With the initial weight vectors in each class computed by adding random vectors of the class variance to the mean vector of the same class, the training process starts from a good statistical mapping position.

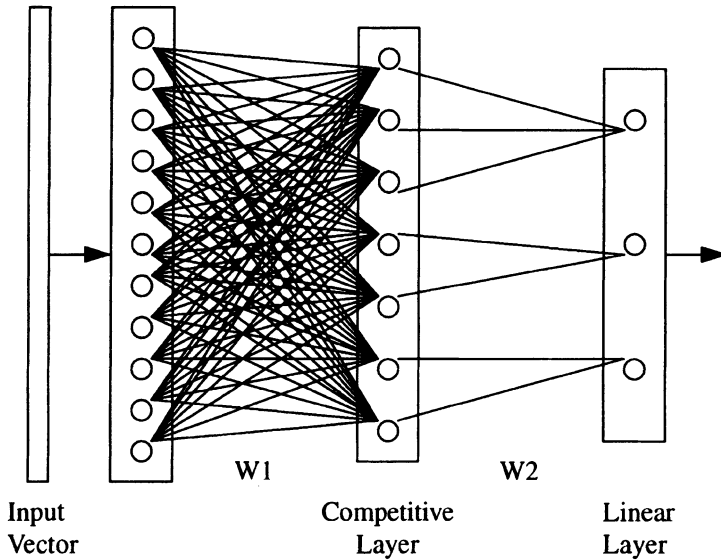


Figure 4. Learning vector quantization neural network.

We make a second improvement by invoking the parallel training strategy (Tang 1996). Traditionally, training samples are randomly selected to be fed into the network one at a time. Therefore only one neuron is updated at each training epoch. The process is quite slow, when there are many neurons to update. The fact that the order in which samples are presented to the network is not important suggests the idea of presenting several training samples in parallel. The only difference this may create is that there may be more than one training sample very near the same neuron. In such a case we select only one of these few samples to update the neuron. This guarantees that parallel processing generates only one update operation on each winning neuron, with results similar to a serial training strategy. The parallel processing can be implemented by either hardware matrix operations or multiprocessor technology.

3. Data Acquisition and Processing

Data acquisition and processing for the current implementation are carried out in two phases. First, a real-time hardware/software system detects in-focus objects, defines a subimage around each detected object, then saves the subimages to disk. Digital storage requirements are reduced by more than two orders of magnitude and the time required to manually identify training images is

accelerated by a similar factor, when compared with manually jogging a videotape editing machine to search for organisms (Davis et al. 1996). In a second phase, more computationally expensive algorithms are applied to the condensed data set for segmentation, feature extraction, and classification.

3.1. *Real-time data acquisition and focused object detection*

The VPR uses a video camera with telephoto lens and a red strobe to obtain magnified images of plankton. The strobe is synchronized with the camera at 60 fields per second. Together, the camera's high resolution (570×485 pixels) and the strobe's short pulse duration allow detailed imaging of the plankton ($10\text{-}\mu\text{m}$ resolution for the 0.5-cm field of view) (Davis et al. 1992). The goal of the VPR's real-time video processing system is to archive to tape digital images of sufficiently large, bright, and in-focus objects as they appear in the video stream.

Live or recorded video and time-code data are sent to the video processing system, which consists of a Sun SPARCstation 20/72 connected to an Imaging Technologies 151 pipelined image processor and a Horita time-code reader (Davis et al. 1996). The image processor can perform real-time (60 field per second) digitization, 3×3 convolutions, rank value filtering, and frame buffer data exchanges with the host workstation. A multi-threaded algorithm on the host is used to supervise the image processor, collect time-code data, compute edge strength, and transfer in-focus subimages to disk.

A simple but effective algorithm has been devised to detect and record objects in real-time (Davis et al. 1996). First, bright large blobs are located in a median filtered image by finding the connected components from run-length lists computed in hardware. For each large blob, the first derivative of the Sobel edge intensity (basically a second derivative of intensity) along the blobs' perimeter is used to reject objects that are out of focus. A gray-scale subimage surrounding any in-focus targets is immediately passed to the workstation for archival and taxonomic classification.

The three main parameters of these algorithms are image intensity threshold, edge strength threshold, and minimum object area in pixels. These are adjusted empirically before data collection based on various factors including lens magnification, transmissivity of the water, and lighting characteristics. Our objective is to achieve a very high probability of detection with a reasonable probability of false alarm, since more intensive processing can be applied to the much smaller data set produced by these algorithms.

On average, about 1 out of 20–60 video fields contains an in-focus object, and only a subimage surrounding the object is saved to disk as an individual file. These object files are time-stamped using the corresponding video time code for precise correlation with ancillary hydrographic and position data

(Davis et al. 1992). This culling process typically reduces the amount of image data to be stored and classified by a factor of 100 or more, thus making the remaining processing computationally feasible (Davis et al. 1996).

3.2. *Data description and processing*

In the following experiment, six classes obtained from nearly 2,000 plankton subimages captured by the VPR are used to test our pattern classification algorithms. They include 133 Calanus, 269 Diat-centr, 658 Diat-chaet, 126 Diatom, 641 Diatomcolo, and 42 Pteropod images. Some sample images for each of the six plankton taxa are illustrated in Figure 1. Half of the images are used as training data and half for testing.

Each gray-scale image is first segmented into a binary image using a simple mean shift method. We use the mean value of the image to threshold the image, then the mean values of the object and the background are computed. The average of the two mean values is used as a new threshold to segment the image again. The process iterates until a stable threshold is reached. Since our images are mostly bimodal, only two iterations give a very good segmentation result, as shown in Figure 5(b). We are currently developing a more robust connectivity-based thresholding technique and will compare the two methods in a future work.

Next, the largest binary object is used to compute the boundary descriptors. This binary image is also used to mask the original grey-scale image to compute moment features. Results for each of these processing steps are illustrated in Figure 5. Granulometric features are computed directly from the original gray-scale images.

4. **Classification Experiments**

A set of experiments was conducted to study the performance of the three types of feature vectors, their combinations, and the improved LVQ classifier. Since there seems to be no simple solution for determining the best network configuration, no exhaustive search was conducted to determine the best network parameters. However, we investigated several network configurations and selected one that appears to be most appropriate for our application. Throughout the experiment, we use the following parameters: 200 competitive layer neurons, learning rate 0.1, parallel training sample number 120 per epoch.

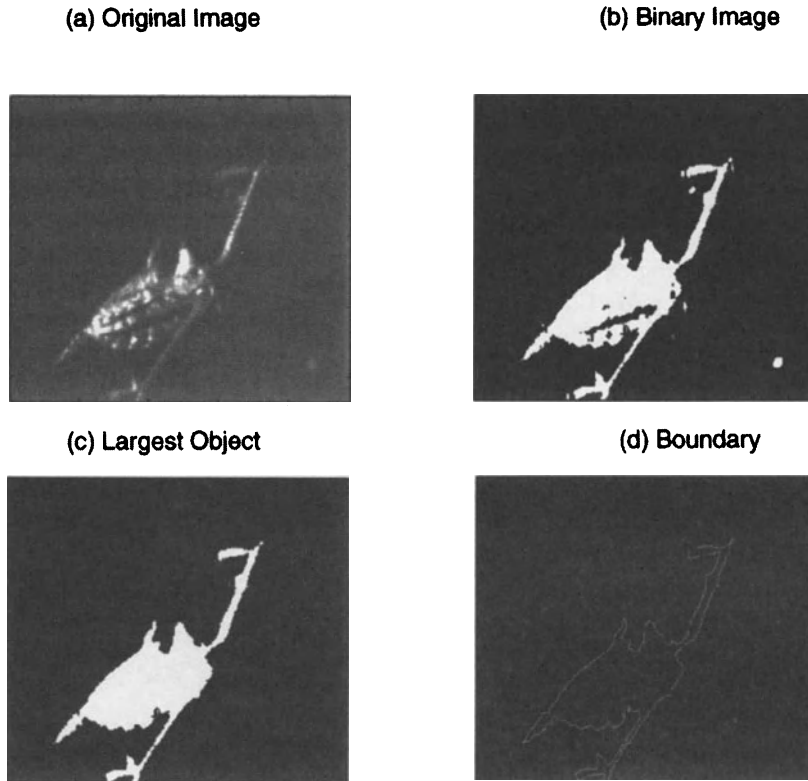


Figure 5. Illustration of the intermediate results of the image processing steps.

4.1. Classification results using individual feature vectors

We first investigate the classification ability of the three individual feature vectors: moment invariants, Fourier descriptors, and granulometries. Comparison of the classification results are given in Table 1. Only 65% classification accuracy is achieved using moment features. This may be attributed in part to the relatively short feature vector length of 7. The images of the same class are also very different in shape because of variations in projection direction, organism body motion, and image occlusions. This shape inhomogeneity also affects the performance of the FD features. For the testing data, only 69% classification accuracy is achieved by the contour FDs. The radius FD features outperform the contour FD features by approximately 10%. This contrasts with Kauppinen et al. (1995), where contour FDs give better results than the radius FDs. Such a discrepancy in results may be caused by the differences between the two data sets. More conclusive comparison studies need to be done before we draw a firm conclusion. The sampling number we used for

Table 1. Classification rates (%) on six classes of plankton images using individual feature vectors: Moment invariants, Fourier descriptors (FD), and Granulometries

Feature types	Original feature length	Number of selected features	Correct classification rate (%)		
			Training	Testing	All data
Moment invariants	7	7	67.74	63.13	65.44
Contour FD	360	28	83.7	69.2	76.5
Radius FD	180	21	94.6	78.0	86.3
Granulometries	160	29	97.8	86.4	92.1

the boundary function is 360 points. This number is much higher than for many previous studies, because plankton have noisier, more irregular boundaries requiring a higher sampling rate to capture high-frequency information. The granulometry features give the best performance with a better than 90% accuracy. This demonstrates the features' insensitivity to occlusion, image projection direction, and body motion because of rich three-dimensional texture and shape information captured.

4.2. Classification results using combined feature vectors

The confusion matrices of the classification results using the three individual features are given in Tables 2–4, where the columns are true class labels, the rows are the resulting labels. Notice that the shapes of these matrices are quite different. The moments are good at distinguishing Diat-centr and Diatom; the radius FD generates good results on Diat-centr and Diat-chaet; the granulometry features perform well on all classes, except on the Diat-centr. All these suggest that the discrimination abilities of the three feature types may be distinct when recognizing different object characteristics. To form a more complete description of the plankton patterns, we combine all three feature vectors into a single feature vector. Results are shown in Table 5. The combined vector yields 95% classification accuracy, which is comparable with what a trained biologist can achieve by using conventional manual techniques (Davis 1982). Notice also that the feature length is condensed to around 20 from several hundreds, demonstrating the efficiency of our feature selection approach.

Not all combinations of feature vectors in Table 5 show an improvement in results. For example the combined contour and radius FDs perform worse than the radius FD features alone. Adding moment features to the granulometry feature vector only improves results slightly. The short moment feature vector seems to contain only a subset of information contained in the granulometry features.

Table 2. Confusion matrix of the moment invariant features

Names	Calanus	Diat-centr	Diat-chaet	Diatom	Diatomcolo	Pteropod
Calanus	41	0	71	0	20	4
Diat-centr	3	241	3	0	65	8
Diat-chaet	67	3	446	8	180	11
Diatom	0	0	10	116	0	0
Diatomcolo	21	23	127	2	373	13
Pteropod	1	2	1	0	3	6

Table 3. Confusion matrix of the radius Fourier descriptors

Names	Calanus	Diat-centr	Diat-chaet	Diatom	Diatomcolo	Pteropod
Calanus	71	0	9	3	26	4
Diat-centr	0	262	2	0	4	1
Diat-chaet	22	6	602	5	66	3
Diatom	2	0	3	117	1	3
Diatomcolo	34	1	42	1	539	9
Pteropod	4	0	0	0	5	22

Table 4. Confusion matrix of the granulometry features

Names	Calanus	Diat-centr	Diat-chaet	Diatom	Diatomcolo	Pteropod
Calanus	117	2	15	2	7	0
Diat-centr	0	239	7	0	0	0
Diat-chaet	15	27	600	5	25	0
Diatom	1	0	16	116	2	0
Diatomcolo	0	1	20	3	607	0
Pteropod	0	0	0	0	0	42

Table 5. Classification rates (%) on the six classes of plankton images using combined feature vectors

Feature types	Original feature length	Number of selected features	Correct classification rate (%)		
			Training	Testing	All data
Contour FD & Radius FD	540	29	90.4	76.8	83.6
Moments & granulometry	167	29	97.5	87.5	92.5
Granulometry & radius FD	340	24	98.7	91.5	95.1
Moments & granulometry & radius FD	347	19	98.0	92.2	95.1

Table 6. Confusion matrix of the combined moments, radius FDs, and granulometry features

Names	Calanus	Diat-centr	Diat-chaet	Diatom	Diatomcolo	Pteropod
Calanus	120	0	9	1	8	1
Diat-centr	0	262	7	0	1	2
Diat-chaet	13	6	627	2	23	0
Diatom	0	0	5	120	0	0
Diatomcolo	0	1	10	3	609	0
Pteropod	0	0	0	0	0	39

The combined feature vector confusion matrix is shown in Table 6. Although the DIAT-CHAET images have a classification accuracy of 95.3%, about the average of all classes, many other class images are misclassified as DIAT-CHAET, such as all misclassified CALANUS, most misclassified DIAT-CENTR and DIATOMCOLO images. This is probably because the versatile DIAT-CHAET images have texture and shape structures similar to those of other class images. For example, the body texture of some DIAT-CHAET look quite similar to that of CALANUS, and some small samples of DIAT-CHAET may be confused with DIAT-CENTR. All images are sorted by a trained biologist and sometimes only very subtle characteristics are used to judge the occluded plankton images. Some human errors in the identification were discovered using the automated method. Given the data quality, the overall classification rate is very encouraging.

4.3. Comparison of LVQ training methods

A traditional serial training method is compared to the new parallel algorithm in Figure 6. The three lines show the progression of training data classification accuracy with increasing number of training epochs using three methods: the traditional training method with all neurons initialized with the mean value of the training samples, the traditional training method with the statistical initial condition, and the new parallel training method. The new method reaches 98% training accuracy within 100 epochs, while the traditional methods achieve 95% accuracy using nearly 10000 epochs.

To further compare the traditional method with the new parallel training method, we use scatter plots of the first two dimensions of the training sample feature vectors and their corresponding neuron weight vectors at several training stages in Figure 7. As the training progresses, the neuron weight vectors gradually start to match the topology of the feature vector space. Comparing Figures 7(a) and (b), we see that the new method apparently maps the density of the feature vector space better and more quickly.

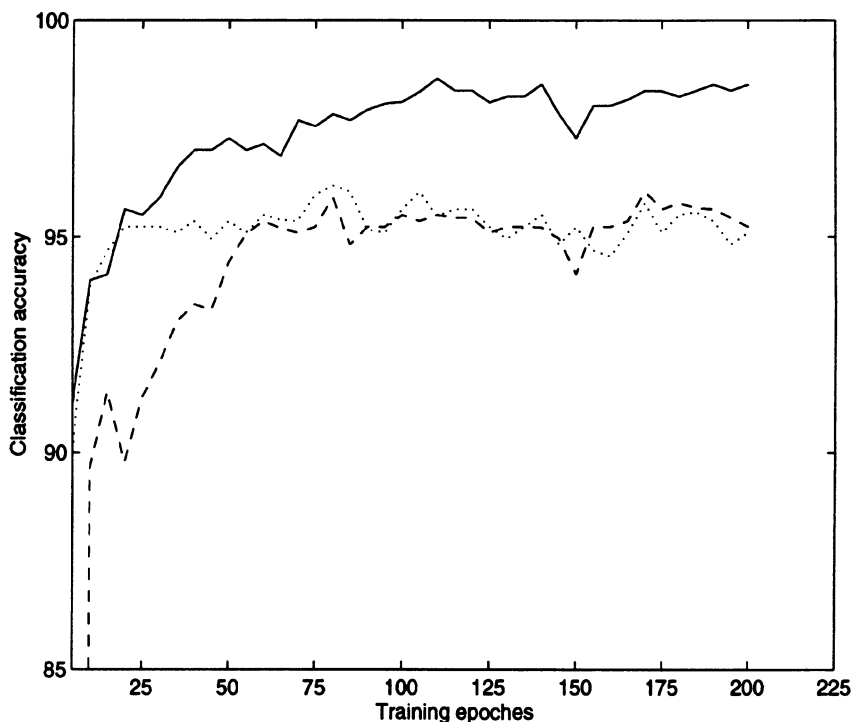
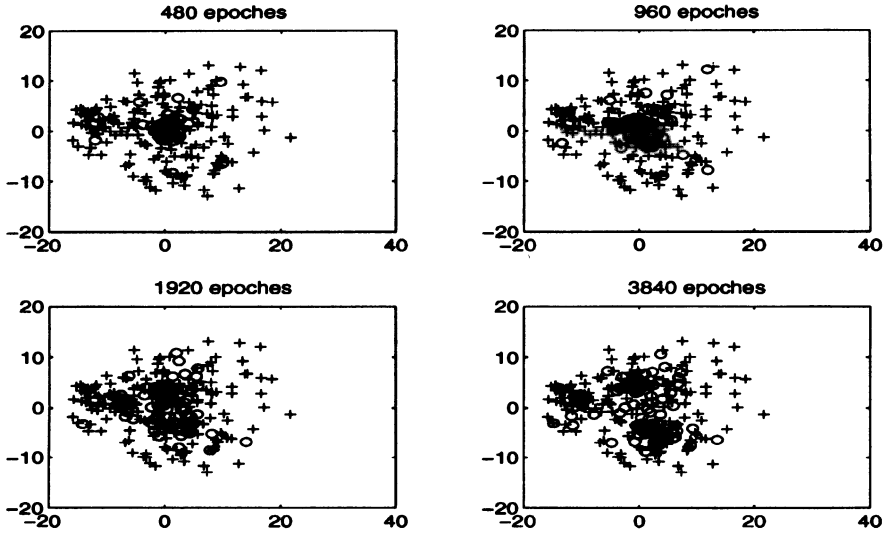


Figure 6. Comparison of the three training methods. The dashed line represents the classification accuracy progression with training epochs using the traditional training method and the mean value initial condition. The dotted line is for the traditional training method with statistical initial condition. The solid line is the result of the new parallel training method. The epoch numbers on the horizontal axis for the two traditional training methods should be multiplied by 120.

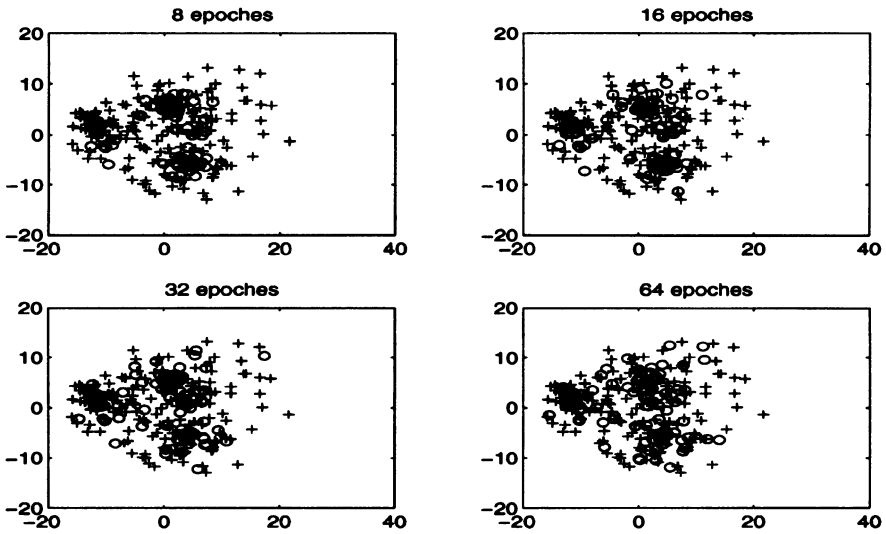
5. Conclusions and Future Work

Our experimental results clearly demonstrate that the combined feature vector is better than any single feature type. Of the individual feature types, the granulometry vector contains more information than conventional shape descriptors. The KLT and Bhattacharyya feature selection method successfully decorrelates and compacts a large feature vector into a small description vector. Together with the improved LVQ classifier, 95% classification accuracy is achieved.

With the VPR system, including the data acquisition unit and the pattern classification system, real-time automatic sorting of plankton into taxonomic categories becomes possible for the first time. This ability will allow rapid acquisition of size-dependent taxonomic data over a wide range of scales



(a) Training using traditional method.



(b) Parallel training with statistical initial condition.

Figure 7. Comparison of the traditional and the new parallel training methods using scatter plots of the first two feature dimensions and their corresponding neuron weight vectors at several training stages. The '+' markers represent the six classes of training samples and the 'o' markers represent the hidden layer neurons.

in a dynamic oceanographic environment, providing new insights into the biological and physical processes controlling plankton populations in the sea (Davis et al. 1992).

To achieve this goal, though, much work remains. First, biologists are interested in classifying many more plankton species than tested here, many of which are quite similar. To maintain a high degree of classification accuracy, we must develop more distinct pattern features. A hierarchical classification system may also be necessary to classify major species and subspecies in several steps. We also intend to integrate our classification algorithms with the current VPR processing system to carry out real-time taxonomic identification at sea.

Acknowledgements

We thank P. H. Wiebe for his comments. This work is supported by the Office of Naval Research through grant N00014-93-1-0602. This is contribution number 9214 of the Woods Hole Oceanographic Institution.

References

- Davis, C. S. (1982). *Processes Controlling Zooplankton Abundance on Georges Bank*. Ph.D. Thesis, Boston University.
- Davis, C. S., Gallager, S. M., Berman, N. S., Haury, L. R. & Strickler, J. R. (1992). The Video Plankton Recorder (VPR): Design and Initial Results. *Arch. Hydrobiol. Beith.* **36**: 67–81.
- Davis, C. S., Gallager, S. M. & Solow, A. R. (1992). Microaggregations of Oceanic Plankton Observed by Towed Video Microscopy. *Science* **257** (10 July): 230–232.
- Davis, C. S., Gallager, S. M., Marra, M. & Stewart, W. K. (1996). Rapid Visualization of Plankton Abundance and Taxonomic Composition Using the Video Plankton Recorder. *Deep-Sea Research II*, Vol. 43, No. 7–8, pp. 1947–1970.
- Fukunaga, K. (1972). *Introduction to Statistical Pattern Recognition*. Academic Press: New York.
- Gonzalez, R. C. & Wintz, P. (1987). *Digital Image Processing*. Addison-Wesley.
- Hu, M. K. (1962). Visual Pattern Recognition by Moment Invariants. *IRE Trans. Information Theory* **IT-8** (Feb.): 179–187.
- Kauppinen, H., Seppanen, T. & Pietikainen, M. (1995). An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification. *IEEE Trans. on Pattern Analysis and Mach. Intell.* **17**(2) (Feb.): 201–207.
- Kohonen, T. (1987). *Self-Organization and Associative Memory*, 2nd Edition. Springer-Verlag: Berlin.
- Kohonen, T. (1990). The Self-Organizing Map. *Proceedings of the IEEE* **78**(9) (Sept.): 1464–1480.
- Maragos, P. (1989). Pattern Spectrum and Multiscale Shape Representation. *IEEE Trans. on Pattern Analysis and Mach. Intell.* **11**(7) (July): 701–716.
- Matheron, G. (1975). *Random Sets and Integral Geometry*. John Wiley and Sons: New York.
- Owen, R. W. (1989). Microscale and Finescale Variations of Small Plankton in Coastal and Pelagic Environments. *Journal of Marine Research* **47**: 197–240.

- Paffenhofer, G. A., Stewart, T. B. Youngbluth, M. J. & Bailey, T. G. (1991). High-Resolution Vertical Profiles of Pelagic Tunicates. *Journal of Plankton Research* **13**(5): 971–981.
- Persoon, E. & Fu, K. S. (1977). Shape Discrimination Using Fourier Descriptors. *IEEE Trans. on Sys., Man, and Cyber* **7**(3) (Mar.): 170–179.
- Reeves, A. P., Prokop, R. J., Andrews, S. E. & Kuhl, F. P. (1988). Three-Dimensional Shape Analysis Using Moments and Fourier Descriptors. *IEEE Trans. on Pattern Analysis and Mach. Intell.* **10**(6) (Nov.): 937–943.
- Reiss, T. H. (1991). The Revised Fundamental Theorem of Moment Invariants. *IEEE Trans. on Pattern Analysis and Mach. Intell.* **13**(8) (Aug.): 830–834.
- Reti, T. & Czinege, I. (1989). Shape Characterization of Particles via Generalized Fourier Analysis. *Journal of Microscopy* **156**(Pt. 1) (Oct.): 15–32.
- Schmitt, M. & Mattioli, J. (1991). Shape Recognition Combining Mathematical Morphology and Neural Networks. *SPIE: Application of Artificial Neural Network*. Orlando, Florida (April).
- Serra, J. (1982). *Image Analysis and Mathematical Morphology*. Academic Press: London.
- Tang, X. (1996). *Transform Texture Classification*. Ph.D. Thesis, Massachusetts Institute of Technology, the MIT/Woods Hole Oceanographic Institution Joint Program.
- Teh, C. H. & Chin, R. T. (1991). On Image Analysis by the Methods of Moments. *IEEE Trans. on Pattern Analysis and Mach. Intell.* **10**(4) (July): 496–513.
- Vincent, L. (1994a). Fast Opening Functions and Morphological Granulometries. *Proc. Image Algebra and Morphological Image Processing*, SPIE 2300, 253–267. San Diego (July).
- Vincent, L. (1994b). Fast Grayscale Granulometry Algorithms. *EURASIP Workshop ISMM '94, Mathematical Morphology and Its Applications to Image Processing*, 265–272. France (Sept.).
- Zahn, C. & Roskies, R. Z. (1972). Fourier Descriptors for Plane Closed Curves. *IEEE Trans. on Computers* **21**(3) (Mar.): 269–281.

Identification and Measurement of Convolutions in Cotton Fiber Using Image Analysis*

YOUNG J. HAN¹, YONG-JIN CHO², WADE E. LAMBERT¹ and CHARLES K. BRAGG³

¹*Department of Agricultural and Biological Engineering, Clemson University, Clemson, SC, USA; (E-mail: young.han@ces.clemson.edu)*

²*Korea Food Research Institute, Songnam-si, Korea;*

³*Cotton Quality Research Station, Clemson, SC, USA*

Abstract. An image analysis procedure was developed to quantify morphological characteristics of convolutions in individual cotton fibers without pre-tensioning or orientation requirements. The image of each fiber was captured by a PC-based color imaging system using a conventional microscope. Ends of individual cotton fibers were glued on a microscope slide without any tension or straightening. A modified watershed technique was implemented to identify individual convolution segments, which were defined as sections of the fiber bordered by two neighboring convolutions. Length, area and perimeter of each convolution segment were measured directly from the image. Average width, shape factor and number of convolution segments in mm were calculated from the measured parameters. Performance of the image analysis algorithm was compared with visual inspection for number and position of convolution segments in three different varieties of cotton. Image analysis results agreed with visual inspection in 89.6% of the tested images.

Key words: convolution, cotton fiber, image analysis

1. Introduction

There have been several studies on characterizing convolutions of cotton fibers and relating convolution characteristics to fiber quality. As early as 1923, Denham (1923) suggested that convolutions might exercise a profound influence on the spinning qualities of fibers. Clegg and Harland (1924) counted the number of convolutions and reversals using a microscope. Meredith (1951, 1953) defined the convolution angle, θ' , as $\theta' = \tan^{-1}(\pi/2)(D/c)$, where D is the ribbon width and c is the pitch of the convolution, and found that the convolution angle and the Pressley strength were highly correlated. Betrabet et al. (1963) reported that the fiber bundle strength was highly correlated with the fibrillar orientation. Hebert (1975) and Duckett (1977) also confirmed that fiber tenacity and convolution angle had high negative correlation.

Convolutions can be measured by optical microscopic method (Clegg 1924; Meredith 1951). Duckett and Cheng (1972) introduced a technique to detect cotton fiber convolutions by a device which allowed azimuthal monitoring of reflected light. They reported that the new technique was as simple as conventional X-ray orientation techniques and faster than microscope techniques. All these methods were time-consuming and laborious to collect individual convolution data.

Thibodeaux and Evans (1986) used image analysis to measure cotton fiber maturity by measuring convolution characteristics. Individual cotton fibers were arrayed longitudinally by pulling them straight and gluing the ends to a standard microscope slide. Although scanning fibers longitudinally is superior to the cross-sectioning approach because of its higher resolution and reduced chance of introducing various errors, pulling the cotton fiber straight may change the shape of a convolution and possibly cause a partial or complete elimination of a convolution. Xu et al. (1992, 1993), on the other hand, asserted that it is more accurate to examine cotton lumens and wall thicknesses from cross-sectional views than from longitudinal view, and developed image analysis techniques for fiber cross-sectional shape analysis to quantify maturity and related characteristics.

Deussen (1990) pointed out that today's high speed and automated cotton processing equipment requires knowledge of the individual and collective effects of raw-material properties on processing behavior. Although HVI systems and conventional laboratory methods are being used for bundle tests, the textile industry can benefit from the study of single fiber properties as well as the data from bundle tests. Single fiber testing, however, is time-consuming and laborious. Therefore, an effective tool for automatic measurement of single fiber properties with simple sample preparation is needed.

The purpose of this study was to develop an image analysis technique to quantify morphological characteristics of convolutions in a single cotton fiber without pre-tensioning or orientation requirements.

2. Imaging Equipment

The image processing hardware used in this study included a MATROX IMAGE-1280 imaging board with IMAGE-CLD color digitizer module from Matrox Electronics Systems Ltd., installed on a 50 MHz 80486 micro-computer. The imaging board had a 1280×1024 spatial resolution and three 8-bit RGB digitizers. Color images of cotton fibers were captured by a SONY DXC 930 color camera with a three-chip CCD image sensor, which was mounted on a Zeiss microscope with $10\times$ objective lens through LA-CS50 optical relay from Century Precision Industries, Inc. The ends of individual

cotton fibers were glued to a conventional microscope slide without any tensioning and illuminated with the standard back-light of the microscope. Color images of cotton fibers were captured at a 512×480 spatial resolution and displayed on a Sony Multiscan 17se high-resolution RGB monitor. The image analysis algorithms developed in this study were programmed in the C Interpreter of Visilog 4.13 from Noesis Vision, Inc..

3. Image Analysis

3.1 *Definition of convolution segment*

Convolution in a cotton fiber is developed during the collapse of the tubular fiber as it dries. Meredith (1951) defined a convolution as a twist of the fiber through 180° about its axis. The longitudinal image of a convoluted fiber resembles a twisted ribbon with a series of bobbin-like segments, where the shape of the bobbin-like segments is affected by how much the fiber is convoluted. A convolution is defined as a border between two bobbin-like segments. In this study, the bobbin-like segment which was bordered by two neighboring convolutions was defined as a *convolution segment*. According to this definition, a convoluted fiber can be considered to be a chain of convolution segments.

Traditionally, convolutions have been characterized by the number of convolutions per unit length, convolution angle, and convolution pitch which is the distance between two neighboring convolutions (Clegg 1924; Meredith 1951; Bertrabet 1963). In this study, convolutions were characterized by the number of convolution segments per mm, and the size and shape of each convolution segment.

3.2 *Image preprocessing*

The image preprocessing procedure consisted of image segmentation, separation of convolution segments by a watershed technique and a skeleton operation. The color image of a cotton fiber had a spatial resolution of 512×480 with 256 intensity levels for each of the red, green and blue components. This color image was converted to a binary image by an automatic thresholding technique. Intensity of each pixel in the input color image with red, green and blue components that satisfy the following relationships were set to 1 (white) in the output binary image:

$$m_R - 3\sigma_R - 0.5 \leq I_R \leq m_R + 3\sigma_R + 0.5$$

$$m_G - 3\sigma_G - 0.5 \leq I_G \leq m_G + 3\sigma_G + 0.5$$

$$m_B - 3\sigma_B - 0.5 \leq I_B \leq m_B + 3\sigma_B + 0.5$$

where,

I = pixel intensity of each color component

m = mean intensity value of each color component

σ = standard deviation of intensity of each color component

Subscript R, G, B: red, green and blue components, respectively

All the other pixels were set to 0 (black) in the output binary image. After this operation, the image of a cotton fiber was shown as white on a black background. It should be noted that a black-and-white imaging system with simple monochrome thresholding would have been just as effective in achieving segmented binary images, because all subsequent processing was done in black-and-white. To clean background noise in the image and to eliminate artifacts inside the fiber, closing and opening operations were applied next. The closing operation fills small holes presented inside an object, connects proximate particles, and eliminates small details by smoothing the boundary from the outside. The opening operation removes small particles and smoothes small details on the boundary such as peaks and isthmuses.

After the binary segmentation, a modified watershed technique (NOESIS 1991; Vincent 1991) was used to separate individual convolution segments in a cotton fiber. The watershed technique is analogous to finding crest lines dividing two disconnected basins in a mountain range. Figure 1 shows the separation of a cotton fiber into convolution segments by the watershed technique. Figure 1(a) shows the longitudinal projection of a typical cotton fiber. Distance from the center of the fiber to the fiber boundary is plotted in Figure 1(b). This extra step in software eliminates the physical orientation requirement that the fiber has to be absolutely straight. Without this extra step, the fiber must be pulled straight before mounting on the microscope slide. Tension in the fiber may change the size and shape of the convolutions.

The next step was to find local minima in the fiber. First, each pixel in the image was assigned a grey level equal to the distance in number of pixels from the boundary of the fiber. Successive erosion operations were performed on this grey level image until only the points with the highest grey level value in the regions remained. The remaining points were mapped vertically down onto the boundary of the fiber to find local minima. To eliminate excessive minima due to irregular shape of the fiber boundary, such as the region between watersheds B and C in Figure 1(b), the points with the highest two grey levels in each region were merged into one modified minimum. The

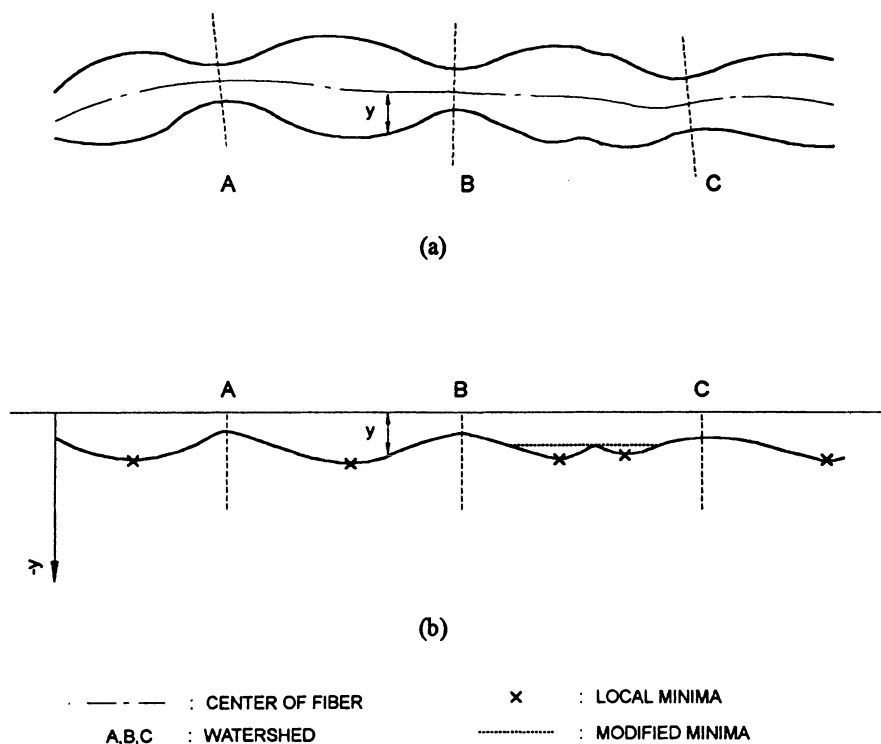


Figure 1. Identification of convolution segments by watershed technique. (a) longitudinal projection of a cotton fiber; (b) distance from the center to fiber boundary.

crest lines, or watersheds, were then found by a simulation of progressive flooding from the modified minima. The watershed was defined as the point (or lines) where two separate basins join as the water rises. In the cotton fiber, watersheds denoted convolutions, and the region between two convolutions was defined as a convolution segment.

Figure 2 shows the result of the modified watershed algorithm. Figure 2(a) shows the original cotton fiber. Different shades in Figure 2(b) show the identified convolution segments. In Figure 2(c), the two segments at either end of the fiber were removed to exclude incomplete segments, and only the remaining whole segments in the figure were used in subsequent analysis. Since the lengths of the curved segments cannot be measured directly from this image, a skeleton of the cotton fiber was obtained by successively thinning and pruning the image into one-pixel-thick line segments as shown in Figure 2(d). Lengths were then measured as one-half of the perimeter of each line segment.

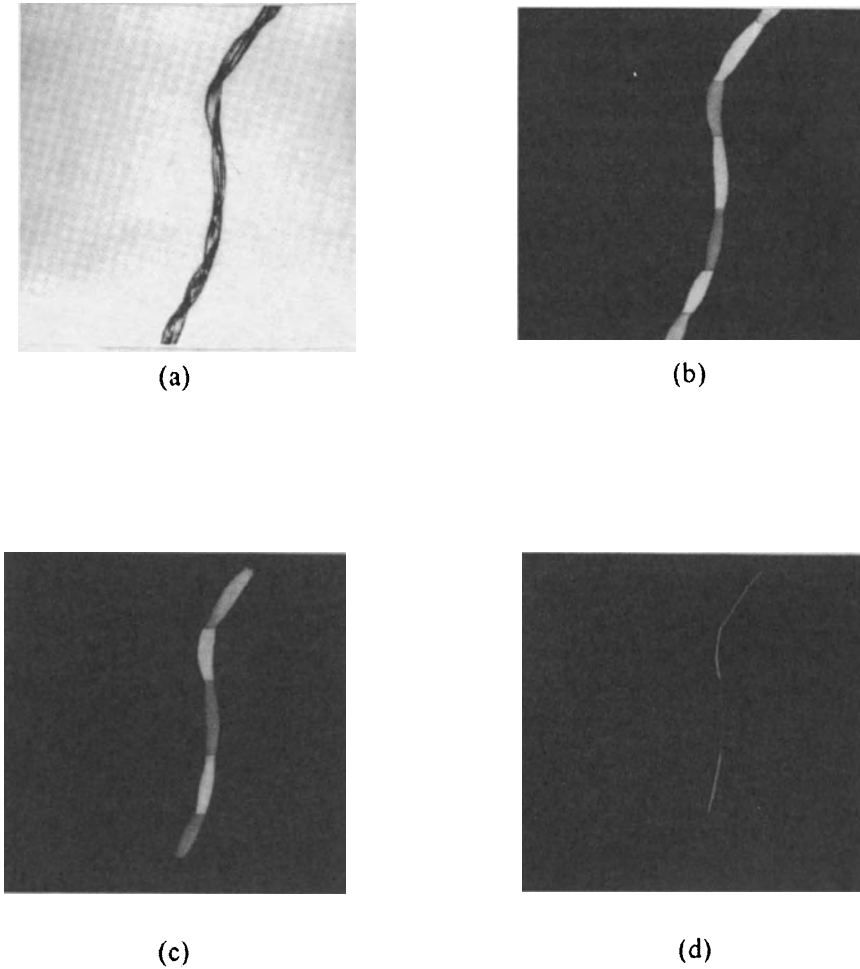


Figure 2. Images of cotton fiber processed by watershed technique. (a) original image; (b) after segmentation and identification of convolution segments; (c) after two end segments removed; (d) skeleton of the fiber for length measurement.

3.3 Image Analysis and Measurements

To characterize convolution segments, the area and perimeter of each convolution segment were measured from the image in Figure 2(c). The area of each segment was measured by counting the number of pixels belonging to each segment and multiplying by scale factors. The horizontal and vertical scale factors were calibrated using a graduated reticle before the measurement

(1.09 $\mu\text{m}/\text{pixel}$ and 0.88 $\mu\text{m}/\text{pixel}$, respectively). Length and perimeter were measured by the Crofton formula (NOESIS 1991), which counts the intercept numbers along major directions and corrects for non-square pixels. Length of a convolution segment means the pitch of the convolution in the traditional definition.

Number of convolution segments per millimeter, average width and the shape factor of each segment were derived from the area, perimeter and length as follows

$$N = 1000/L_a \quad (1)$$

$$W_a = A/L \quad (2)$$

$$F_s = P^2/(4\pi A) \quad (3)$$

where,

N = number of convolution segments per millimeter

L_a = average length of convolution segments in a fiber (μm)

W_a = average width of a convolution segment (μm)

A = area of a convolution segment (μm^2)

L = length of a convolution segment (μm)

F_s = shape factor

P = perimeter of a convolution segment (μm)

The shape factor, F_s , measures the elongation of a convolution segment. The shape factor is equal to 1 for a circle. An elongated segment has a high shape factor. The shape factor has a nonlinear negative relationship with the convolution angle as defined by Meredith – the larger the shape factor, the smaller the convolution angle. The maximum width of a convolution segment, W_m , can be estimated as follows, if a convolution segment is assumed to be an ellipse:

$$W_m = 4A/\pi L \quad (4)$$

4. Validation Test

Three different varieties of cotton; DPL ACALA 90, ACALA MAXXA SJV and PAYMASTER HS 26 were used for validation testing. Ninety cotton fibers, thirty fibers from each variety, were randomly selected. Each single fiber was mounted on a microscope slide glass to capture a longitudinal image of the cotton fiber. Because pulling the fiber may change the shape of convolutions, the ends of the fiber were glued to the slide without tension. Three images were captured from each fiber for convolution measurements (from the

Table 1. Results of convolution recognition by the watershed technique

Variety	Correct Recognition	Counting Error	Positioning Error	Total
DPL ACALA 90	79	10	1	90
ACALA MAXXA SJV	81	7	2	90
PAYMASTER HS 26	82	6	2	90
Total	242	23	5	270

Table 2. Distribution of counting errors by the watershed technique

Variety	Differences in number of convolution*					Total
	-4	-3	-2	-1	+1	
DPL ACALA 90	0	1	1	7	1	10
ACALA MAXXA SJV	0	0	1	6	0	7
PAYMASTER HS 26	1	0	1	3	1	6
Total	1	1	3	16	2	23

*Negative and positive numbers designate numbers of undercounted and overcounted convolutions in each image, respectively.

bottom, middle and top parts of each fiber). The modified watershed algorithm was applied to each image to identify convolution sections. Performance of the image analysis algorithm was compared with visual inspection for number and position of convolution segments in each image. Visual inspection was performed by a person familiar with convolutions in cotton fiber.

Table 1 shows the results of convolution recognition by the watershed technique, when compared with visual inspection. Among the total of 270 images, 242 images, or 89.6%, were correctly recognized by the watershed technique. Number of convolutions was miscounted in 23 images, or 8.5%. In five images, or 1.9%, the number of convolutions was correctly identified but their positions were different from the location indicated by visual inspection. Variation in recognition accuracy among varieties was not statistically significant.

Table 2 shows the distribution of errors in counting convolutions by the watershed technique. Among the 23 miscounted images, 18 images had an error of one convolution overcounted or undercounted. Only two images had errors of more than three convolutions. For some of these miscounted convolutions, collapse of the fiber was hardly visible and ambiguous. If a counting error of one convolution were considered acceptable, it could be

Table 3. Average morphological properties of convolution segments in the tested cotton fibers

Parameter	Variety					
	DPL ACALA 90		ACALA MAXXA SJV		PAYMASTER HS 26	
Length, μm	108.5 a [†]	(26.6) [‡]	107.2 a	(17.6)	90.6 b	(18.2)
Area, μm^2	1850.3 ab	(481.4)	1905.9 a	(400.0)	1642.1 b	(399.4)
Perimeter, μm	255.9 a	(54.7)	254.9 a	(38.0)	220.5 b	(39.3)
Average width, μm	16.9 b	(1.3)	17.6 ba	(1.6)	18.0 a	(2.3)
Maximum width, μm	21.7 b	(1.8)	22.5 ba	(2.0)	23.1 a	(2.9)
Shape Factor	2.86 a	(0.53)	2.76 a	(0.38)	2.41 b	(0.40)
No. of convolution segments per mm	9.66 b	(2.08)	9.58 b	(1.62)	11.43 a	(2.04)

[†] Averages followed by the same letter are not significantly different at 95% confidence level using Duncan's Multiple Range Test.

[‡] Values in parenthesis indicate standard deviations.

said that the watershed technique successfully identified 260 images out of 270 total images with a success ratio of 96.3%.

Table 3 shows the morphological properties of convolution segments in the tested cotton fibers as measured by the image analysis algorithm developed in this study. Large deviations among the tested fibers were observed. For example, the length of convolution segments ranged from 42.8 to 260.1 μm for DPL ACALA 90, from 47.4 to 291.3 μm for ACALA MAXXA SJV, and from 40.5 to 224.6 μm for PAYMASTER HS 26. Despite the variation within varieties, statistical analysis indicated highly significant differences between varieties for length, perimeter, shape factor and number of convolution segments per mm. Area, average width and maximum width were also significantly different among varieties with 95% confidence.

Table 3 also shows the results of Duncan's multiple range test for each parameter, with respect to variety. The test showed that these three varieties can be classified into two separate categories by length, perimeter, width and the shape of the convolution segments. DPL ACALA 90 and ACALA MAXXA SJV had high shape factors and low number of convolution segments per millimeter, whereas PAYMASTER HS 26 had a low shape factor and high number of convolution segments per millimeter. It should be noted that the number of convolutions per mm was not calculated as the reciprocal of the average length of all convolution segments, but as the average number of convolutions per mm calculated for each fiber.

5. Summary and Conclusion

An image analysis procedure was developed to quantify morphological characteristics of convolutions in a single cotton fiber without pre-tensioning or orientation requirements. The image of each fiber was captured by a PC-based color imaging system using a conventional microscope. Ends of individual cotton fibers were glued on a microscope slide without any tensioning or straightening. The color image was converted to a binary image by an automatic thresholding technique, and a modified watershed technique was implemented to identify individual convolution segments in the fiber. Length, area and perimeter of each convolution segment were measured directly from the image. Average width, shape factor and number of convolution segments in mm were calculated from the measured parameters.

The performance of the image analysis algorithm was compared with visual inspection for number and position of convolution segments in three different varieties of cotton. Among the total of 270 images, 242 images, or 89.6%, were correctly recognized by the watershed technique. Number of convolutions in 260 images, or 96.3%, were within one convolution of the visual inspection. In five images, or 1.9%, the number of convolutions was correctly identified, but their position was different from that indicated by visual inspection. Variation in recognition accuracy among varieties was not statistically significant.

Morphological analysis showed that deviations among the tested fibers were much greater than the differences between varieties. Despite the variations within the varieties, statistical analysis indicated highly significant differences between varieties for length, perimeter, shape factor and number of convolution segments per mm. Duncan's multiple range test showed that the three varieties can be classified into two separate categories by length, perimeter, width and shape of the convolution segments. DPL ACALA 90 and ACALA MAXXA SJV had high shape factors and a low number of convolution segments per millimeter, whereas PAYMASTER HS 26 had a low shape factor and a high number of convolution segments per millimeter.

Note

* Technical Contribution No. 4154 of the South Carolina Agricultural Experiment Station, Clemson University.

Mention of specific products is for information only and not to the exclusion of others that may be suitable.

References

- Betrabet, S. M., Pillay, K. P. R. & Iyengar, R. L. N. (1963). Structural properties of cotton fibers. *Textile Res. J.* **33**: 720–727.
- Clegg, G. G. & Harland, S. C. (1924). A study of convolutions in the cotton hair. *The Journal of the Textile Institute* **15**: T14–T26.
- Denham, H. J. (1923). The structure of the cotton hair and its botanical aspects: the morphology of the wall. *The Journal of the Textile Institute* **14**: T86–T113.
- Deussen, H. (1990). Cotton for high-tech spinning: what does the spinner expect from HVI? *Proceedings of 1990 Beltwide Cotton Production Research Conference* **1**: 558–560.
- Duckett, K. E. (1977). Cotton fiber convolutions – their measurement and relation to fiber tenacity. Tennessee Farm and Home Science Progress Report No. 102, 2–5.
- Duckett, K. E. & Cheng, C. C. (1972). The detection of cotton fiber convolutions by the reflection of light. *Textile Res. J.* **42**: 263–268.
- Hebert, J. J. (1975). Effect of convolution angle upon cotton fiber strength. *Textile Res. J.* **45**: 356–357.
- Meredith, R. (1951). Cotton fibre tensile strength and x-ray orientation. *The Journal of the Textile Institute* **42**: T291–T299.
- Meredith, R. (1953). Measurements of orientation in cotton fibres using polarized light. *British Journal of Applied Physics* **4**: 369–373.
- NOESIS (1990). *A tutorial on image processing*, Noesis Vision Inc., Quebec.
- Thibodeaux, D. P. & Evans, J. P. (1986). Cotton fiber maturity by image analysis. *Textile Res. J.* **56**(2): 130–139.
- Vincent, L. & Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transaction on Pattern Analysis and Machine Intelligence* **13**(6): 583–598.
- Xu, B., Pourdeyhimi, B. & Sobus, J. (1992). Characterization of fiber crimp by image analysis: definition, algorithms and techniques. *Textile Res. J.* **62**(2): 73–80.
- Xu, B., Pourdeyhimi, B. & Sobus, J. (1993). Fiber cross-sectional shape analysis using image processing techniques. *Textile Res. J.* **63**(12): 717–730.

Fuzzy Logic for Biological and Agricultural Systems

BRIAN CENTER and BRAHM P. VERMA*

Department of Biological and Agricultural Engineering, University of Georgia, Athens, GA 30602, USA (author for correspondence: E-mail: bverma@bae.uga.edu)*

Abstract. Fuzzy logic is a powerful concept for handling non-linear, time-varying, adaptive systems. It permits the use of linguistic values of variables and imprecise relationships for modeling system behavior. The paper presents an overview of fuzzy logic modeling techniques, its applications to biological and agricultural systems and an example showing the steps of constructing a fuzzy logic model.

Key words: agricultural systems, biological systems, fuzzy logic, modeling

1. Introduction

Zadeh (1965) introduced the concept of fuzzy sets as a means for describing complex systems without the requirements for precision. Zadeh (1973) proclaimed a principle called the “principle of incompatibility”, which states that complexity and precision are incompatible due to the inability of the human mind to comprehend complex systems in a detailed manner. By reducing the need for precision it is possible to more easily express known qualitative relationships about complex systems. Zadeh (1973) noted that this method for dealing with uncertainty would have particular applicability in soft systems such as psychology, sociology, and economics. Fuzzy logic may also be useful for descriptive systems, those that fall somewhere between hard systems and soft systems, such as biology and agriculture.

Following the introduction of the fuzzy set, Zadeh (1973, 1975a, 1975b, 1976) began to develop the necessary inferencing mechanisms and modeling techniques to bring this concept to fruition. Within this same time period Assilian (1975) recognized that fuzzy logic could aid in the development of control systems in which nonlinearities and time-variance make the development of traditional control systems very difficult. In these cases a human operator is capable of controlling the plant, so a fuzzy controller can often be designed based on the operator’s expert knowledge. Since that time, control applications have been extensively investigated (Lee 1990).

Since fuzzy systems are similar to expert systems in their use of linguistic relationships, they can be applied to decision support systems (Turksen et al. 1993). Some examples of their application have been in managing liquidity flows of banks (Gardin et al. 1995), selecting workers for training programs (Ntuen and Chestnut 1995), and rating strategic goals (Lasek 1992).

Modeling of complex systems is the final application for which fuzzy logic has been used. Some examples include modeling the start up cycle of gas turbine engines to aid in the evaluation of performance (Balakrishnan and Santhakumar 1996), predicting the seasonal demand for electric power (Dash et al. 1995), meeting quality control objectives with minimum cost in a wood chip refining process (Qian and Tessier 1995), and analyzing consumer preference data for marketing studies (Yager et al. 1994). Although Zadeh predicted that fuzzy models would find widespread use in soft systems, only preliminary investigations have begun in the social sciences (Fedrizzi et al. 1993; Barbera and Albertos 1994).

The applications of fuzzy logic to the descriptive systems of biology and agriculture are beginning to be investigated as will be discussed in Section 3. The basic components of fuzzy models and techniques that have been used for their development will also be discussed.

2. Components of Fuzzy Models

Modeling with fuzzy logic is analogous to classical modeling. The models have variables which influence system behavior and relationships among the variables which describe the system. In classical models variables have real number values, the relationships are defined in terms of mathematical functions, and the outputs are "crisp" numerical values. In fuzzy logic however, the values of variables are expressed by linguistic terms such as "large, medium, and small", the relationships are defined in terms of if-then rules, and the outputs are fuzzy subsets which can be made "crisp" using defuzzification techniques. The crisp values of system variables are fuzzified to express them in linguistic terms. Fuzzification is a method for determining the degree of membership that a value has to a particular fuzzy set. This is determined by evaluating the membership function of the fuzzy set for the value.

2.1 Membership functions

2.1.1 Definition

A fuzzy subset A is defined by a membership function, $\mu_A(t)$, where t is the domain of the variable on which A is defined. The value of $\mu_A(t)$ for each t determines the degree to which each element in the domain belongs

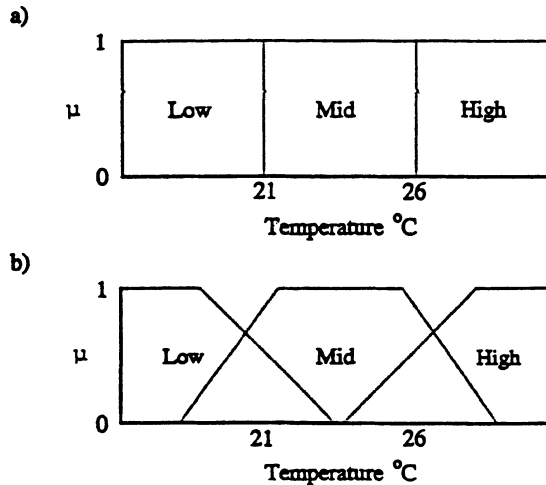


Figure 1. (a) Membership functions for classical sets. Only step functions are allowed as shapes of the membership functions. (b) Membership functions for fuzzy sets. The membership functions are allowed to be any shape. Trapezoidal sets are shown here.

to A. Although both classical and fuzzy subsets are defined by membership functions, the degree to which an element belongs to a classical subset is limited to being either zero or one. This means that $\mu(t)$ may only be a step function. On the other hand, in fuzzy logic the degree to which an element belongs to a subset may be any value in the interval $[0, 1]$. Since $\mu(t)$ for a fuzzy subset may be defined by any function, it is easy to see that a classical subset is a special case of a fuzzy subset (Kosko 1993). Graphically, memberships for classical and fuzzy subsets are shown in Figure 1. The fuzzification process entails inputting the current value of the variable into the membership functions to determine the degree of membership that the value has to each of the subsets defined by the membership functions. More rigorous definitions of membership functions may be found in (Yager and Filev 1994; Gebhardt and Klawonn 1994; Zadeh 1965).

2.1.2 Methods for determining membership functions

One technique for determining the shape of membership functions is seeking knowledge of the system from experts, then constructing membership functions that represent expert opinion. Observations of many experts is preferred as they represent system understanding from different perspectives. From observations of experts, membership functions can be constructed using statistical methods and the analysis of preferences (Kruse et al. 1994). Surveys comparing different approaches for applying this type of analysis can be found in (Turksen 1991; Norwich and Turksen 1984).

Many researchers have investigated more “rational” techniques for determining membership functions. One of these approaches is fuzzy clustering (Sugeno and Yasukawa 1993; Yoshinari et al. 1996; Hwang and Woo 1995). Fuzzy clustering requires a set of input and/or output data from the system to be collected. Clusters of data are found within the input or output space such that the variance within clusters is minimized and the variance between clusters is maximized (Sugeno and Yasukawa 1993). The prototype, or center point, of a cluster X is determined and a distance metric is used to determine the degree of membership that a datum has to X . The membership of an element to X is a normalized measure of the distance to the center. A projection of X onto one of the axes gives the membership function for a subset X_1 on that variable. Similarly, other subsets for the variable can be obtained from projections of other clusters, and subsets for other variables can be obtained by projecting onto the other axes. Excellent exposition on fuzzy clustering can be found in (Yang 1993; Bezdek 1981; Bezdek 1992; Gustafson and Kessel 1984).

Another technique for determining membership functions involves neural networks. Neural networks are networks of simple processors connected by adjustable weighting factors (Kosko 1992). The weights are adjusted using a backpropagation or other known algorithm to minimize the difference between the predicted output and the measured output. The parameters of membership functions can be “learned” by a neural net from a set of input-output data (Takagi and Hayashi 1991; Keller and Hunt 1985; Jang and Sun 1995; Williams 1994).

Genetic algorithms have also been used to determine the optimal shape for membership functions (Wiggins 1992; Karr 1991a; Karr 1991b; Karr and Gentry 1993; Park et al. 1994). Genetic algorithms are a model of evolution in which the population is composed of character strings (Beasley et al. 1993; Goldberg 1989). The parameters of the membership functions are encoded into these strings which may mutate and exchange information. The strings which best meet the objective function reproduce, while the inferior ones are deleted from the population. At the end of the process an optimal set of parameters are expected to “evolve.”

2.2 Fuzzy operators and inferencing

For fuzzy subsets to be useful in modelling, there must be a way to define relationships between them. This is accomplished with logical operators and inferences. Logical operators can form a new subset from two or more existing subsets. An inferencing determines how to construct an output subset from input subsets using relationships expressed by if-then statements.

2.2.1 Operators

There are three basic operators in fuzzy logic, which are analogous to the boolean operators: (1) union or OR, (2) intersection or AND, and (3) negation or NOT. Fuzzy logic also uses the same operators, but differ somewhat from their boolean counterparts due to the necessity of dealing with partial memberships. The most common definition for these three operators are

$$\text{AND: } A \wedge B = \min(\mu_A, \mu_B)$$

$$\text{OR: } A \vee B = \max(\mu_A, \mu_B)$$

$$\text{NOT: } A = 1 - \mu_A$$

The *min* and *max* operators are special cases of t-norms and t-conorms respectively. Other definitions for the union and intersection can be used. A more complete discussion of some of the alternatives can be found in (Yager and Filev 1994).

2.2.2 Inferencing

An implication is an if-then rule, just like the production rules in classical expert systems. The goal is to obtain a conclusion consisting of one or more consequents from a premise consisting of one or more antecedents. A set of such rules may include the following:

IF A is a AND B is b* THEN C is c

IF A is a* AND B is b THEN C is c*

First the truth of each of the premises is evaluated from the current membership of A and B to the subsets a, a*, b, and b*. The activation level of each rule is then equal to that of the premise. So, for example, if A is a AND B is b* evaluates to 0.7 then the activation level of the first rule is 0.7. This determines how to scale or clip the output fuzzy set. In Max-Min inferencing, the output fuzzy set is scaled by the activation level (Tog 1993). In Max-Dot inferencing, the output fuzzy set is clipped at the level of activation (Tog 1993). These are the two most common inferencing techniques used.

Since A may have partial membership to both a and a* and B may have partial membership to both b and b*, both of the rules may have non-zero activation levels. In this case, both rules contribute to the final evaluation for C. The result for C will be a subset comprised of a combination of the output fuzzy sets for each rule. The combination will either be the union of the sets if a constructive solution approach is used, or the intersection, if a destructive approach is used (Yager and Filev 1994).

2.2.3 Defuzzification

The desired output of a fuzzy model is often not the fuzzy subset for C , but a single element within the subset. In order to get a precise value, the output subset is defuzzified. There are many techniques for defuzzification, such as center of area, mean of maxima, basic defuzzification distribution (BADD), semi-linear defuzzification (SLIDE), modified semi-linear defuzzification (M-SLIDE), and random generation (RAGE) (Yager and Filev 1994). All of the techniques try to find the expected value from a probability distribution and differ only in their technique of estimating the probability distribution from the fuzzy subset (Yager and Filev 1994). The most common technique is center of area, in which the centroid of the output subset is the final value. The centroid is the value at which the moments of the sets comprising the output subset sum to zero. The moment is the value at which the area to the left of the value is equal to the area to the right of the value.

2.3 Methods for determining rules

Expert knowledge is the most common technique for determining rules. The expert is asked to summarize the knowledge about the system in the form of cause and effect relationships. From these the rules are formulated. When no experts are available or a more analytical approach is desired, other techniques of system identification must be used.

An approach used is to apply a set of metarules to adaptively acquire information about the system. Linguistic self-organizing controllers issue control actions and observe the environment (Procyk and Mamdani 1979). The metarules evaluate the performance of the control actions from the effect of the control actions on the environment. The metarules then determine if the control actions should be modified and perform the modification if necessary. This technique has been applied to robotics applications (Tansheit and Scharf 1988).

Fuzzy classifier systems are another way to discover rules. Fuzzy classifier systems are generalizations of genetic classifier systems. A genetic classifier system uses some of the ideas from genetic algorithms to develop expert systems (Booker et al. 1989; Goldberg 1989). Since a fuzzy system is an expert system in which the linguistic expressions are given a fuzzy rather than a classical definition, the classifier system may be generalized to fuzzy system generation (Valenzuela 1992). Standard genetic algorithms have also been used to discover rules (Park et al. 1994).

Sugeno and Yasukawa (1993) use fuzzy clustering on data in the output space to determine output fuzzy subsets. The clusters in the output space are used to induce clusters in the input space. Then input space clusters are projected onto the axes to find the fuzzy subsets for the input variables. Since

the input clusters are induced from the output clusters, rules can be constructed between the input subsets and the output subsets. Yoshinari et al. (1996) discuss another method of fuzzy model construction based upon clustering techniques. This method produces a relational as opposed to a functional model, and consists of a series of local, linear models to approximate a global non-linear one. Hwang and Woo (1995) use a hybridized fuzzy clustering and genetic algorithm system to generate fuzzy models. Recently, neural networks have become an active field of interest and have been used to learn rules as well as membership functions (Keller and Tahani 1992; Jang and Sun 1995; Jang 1993).

3. Applications of Fuzzy Logic to Biological and Agricultural Systems

Mechanisms of biochemical reactions and their control in living systems are complex, non-linear, and time-variant (Yamakawa 1992). Furthermore, characteristics of biomass (e.g., a leaf or a fruit) are affected by the environment causing adaptation in the particular set of biochemical reactions used by the system. This adaptation can not be easily described by mathematical equations. In many cases biomass characteristics are not directly measurable and only rough information about the system can be obtained. This is the nature of understanding of living systems at both the cell-level and at the organism-level. Fuzzy logic approaches provide a suitable framework for modeling these systems due to their ability to handle "fragmentary, uncertain, qualitative and blended knowledge typically available for biological systems" (Konstantinov and Yoshida 1992). A brief description of examples of a fuzzy model, a fuzzy decision support system, and fuzzy control systems are presented below.

Center and Verma (1992) reported a fuzzy model of photosynthesis of tomato plants grown in greenhouses by considering ambient temperature, carbon dioxide concentration, photon flux density, distribution of leaves in the plant canopy, and shading effects. The fuzzy model outputs of the total photosynthesis for several plant growth stages and environmental conditions were found to be representative of the experimental field data from Florida.

Verma (1995) developed a fuzzy decision support system (DSS) to aid decisions related to quality sorting of tomatoes. Tomato quality was described as a complex combination of quality characteristics, color (brightness and hue), sensory color, fruit size, shape, and firmness. The expected changes in quality characteristics of fruits with time during storage and transportation conditions are also critical to this problem. The fruit quality was the sole determinant for sorting and the future sell-day was to be predicted when the highest quality was anticipated. Six fuzzy models were developed and linked

to develop the DSS. The outputs of fuzzy DSS predicting quality and the day of the highest quality was very accurate when compared with the data provided by an expert. However, only a limited testing is reported.

The fermentation industry was one of the first to recognize the potential of fuzzy control for biological processes (Konstantinov and Yoshida 1992; Dohnal 1985). Konstantinov and Yoshida (1990a, 1989) have developed a fuzzy logic system for identifying physiological states in fermentation processes to use in distributed control. This system has been applied to phenylaline production in *E. coli* (Konstantinov and Yoshida 1990b). Flanagan (1989) used fuzzy control in sludge processes used in wastewater treatment.

Some other examples of reported fuzzy logic applications includes a model to predict the effects of multiple stresses on tree growth (Schmoldt 1991), organizing bioengineering knowledge in fuzzy models that can be used for prediction (Dohnal 1985; Linko 1988), predicting right soil moisture for land preparation (Thangaradivehu and Colvin 1993), feeding strategies on large dairy farms (Edan et al. 1992), and grading beef quality (Nakamishi et al. 1993). Based on the growing evidence from the literature showing successful use of fuzzy logic for modeling, DSS and controls, it appears that applications to biological and agricultural systems are inevitable.

4. An Example

In this section we will go through an example of the construction and operation of a simplified fuzzy model to predict plant growth as a function of sunlight and soil moisture. The first step is to develop a descriptive understanding of the system and obtain data containing a range of values for sunlight and soil moisture and the resulting growth rate of the plant. From data we can determine the appropriate operational range (universe of discourse) for each variable. Upon the universe of discourse we can define membership functions using the techniques discussed in Section 2.1.2. Figure 2 illustrates assumed membership functions for the selected independent and dependent variables. Let sunlight be 200 PAR and soil moisture be 3 L/m³. The values are fuzzified using the membership functions from Figure 2 as shown by the vertical lines. The value for sunlight belongs to the subset Mid to a degree of 1. The value of soil moisture belongs to the subset Low to a degree of 0.8 and to the subset Mid to a degree of 0.2

Next we determine the relationships between the variables and express them using linguistic rules. The rules can be determined using expert knowledge, clustering, or one of the other techniques discussed in Section 2.3. The results are a set of rules. Figure 3 presents a possible set of rules for the example model presented.

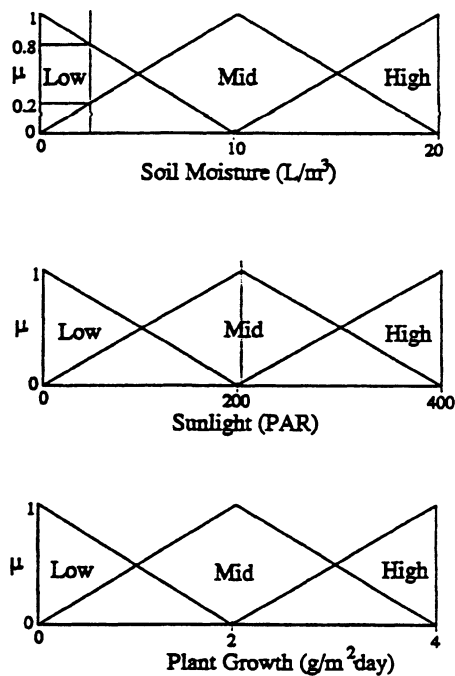


Figure 2. Assumed sets of membership functions for the variables Soil Moisture Sunlight, and Plant Growth.

		Sunlight		
		Low	Mid	High
Soil Moisture	Low	Low	Low	Mid
	Mid	Low	Mid	High
	High	Mid	High	High

Figure 3. Rules which describe the relationships of variables. The value of Plant Growth which results from the combination of Soil Moisture and Sunlight values appears in the cells. For example, the first cell of the table should be read "if sunlight is low and soil moisture is low then plant growth is low."

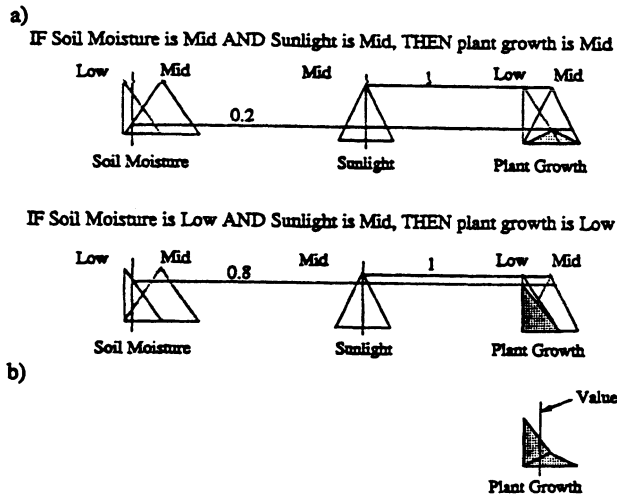


Figure 4. (a) The vertical lines represent the current value of the variables. The minimum membership to the two conditions is taken as the level of activation for the rule. (b) The fuzzy subset which results from the union of all rule outputs. The vertical line represents the centroid of the subset which is the final value for plant growth.

After the values for sunlight and soil moisture are input into the model, the rules are assessed using Max-Min inferencing as shown in Figure 4a. Since $\min(\mu_M, \mu_M) = \min(0.2, 1) = 0.2$, the activation level of the first rule is 0.2 and the plant growth is Mid to a degree of 0.2. Similarly, $\min(\mu_L, \mu_M) = \min(0.8, 1) = 0.8$, so the activation level of the second rule is 0.8 and the plant growth is Low to a degree of 0.8. The resulting fuzzy subset for plant growth is a composite of the output from the two rules as shown in Figure 4b. The defuzzified plant growth value is obtained from the center of area method as shown by the vertical line labeled value in Figure 4b. The shaded area corresponding to the truth value of the first proposition is equal to 0.4 and the moment is equal to 2. The shaded area corresponding to the truth value of the second proposition is 0.8 and its moment is equal to 0.5858. So, the plant growth is predicted to be

$$\frac{\sum_{i=1}^n \mu_i \cdot M_i}{\sum_{i=1}^n \mu_i \cdot A_i} = \frac{0.2 \times 2 + 0.8 \times 0.6}{0.2 \times 0.4 + 0.8 \times 0.8} = 1.2 \text{ g/m}^2 \cdot \text{day}$$

under the given conditions, where M is the moment and A is the area.

To evaluate the model, its output is compared with additional data (test data) collected from the system. It is important that the test data is not used in the construction of the model, so that the predictive capability of the model can be objectively assessed. If the model outputs are not representative of the plant growth as compared against the test data, then the shape of the

membership functions and/or linguistic rules may be altered. These changes are governed by the understanding of the variables and their interactions. Also, such techniques as neural networks or genetic algorithms can be used to modify the membership functions as discussed in Section 2.1.2.

5. Conclusions

Fuzzy logic provides a methodology for describing complex systems using qualitative relationships in a manner analogous to quantitative equations. Non-linear, time-varying, adaptive systems are too complex to be accurately represented using conventional methods. Fuzzy logic has been used to model fermentation process, photosynthesis, food quality, soil properties for trafficability, bioengineering, economics, social behavior, and a variety of other complex systems. Fuzzy models have also been used to solve control problems. This technique is particularly useful for agricultural and food systems which are complex and do not easily lend themselves to analytical approaches. We anticipate a greater use of fuzzy logic for these systems in the future.

References

- Balakrishnan, S. & Santhakumar, S. (1996). Fuzzy modeling considerations in an aero gas-turbine engine start cycle. *Fuzzy Sets and Systems* 78(1): 1–4.
- Barbera, E. & Albertos, P. (1994). Fuzzy-logic modeling of social-behavior. *Cybernetics and Systems* 25(2): 343–358.
- Beasley, D., Bull, D. & Martin, R. (1993). An overview of genetic algorithms: Part 1, fundamentals. *University Computing* 15(2): 58–69.
- Bezdek, J. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press.
- Bezdek, J. (1992). *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data*. New York: IEEE Press.
- Booker, L., Goldberg, D. & Holland, J. (1989). Classifier systems and genetic algorithms. *Artificial Intelligence* 40: 235–282.
- Center, B. & Verma, B. (1992). A fuzzy photosynthesis model for tomato. *Technical Report No. 95-3558*. St. Joseph, MI: ASAE.
- Dash, P., Liew, A., Rahman, S. & Dash, S. (1995). Fuzzy and neuro-fuzzy computing models for electric-load forecasting. *Engineering Applications of Artificial Intelligence* 8(4): 423–433.
- Dohnal, M. (1985). Fuzzy bioengineering models. *Biotechnology and Bioengineering* 27: 1146–1151.
- Edan, Y., Grinspan, P., Maltz, E. & Kahn, H. (1992). Fuzzy logic for applications in the dairy industry. *Technical Report No. 92-3600*. St. Joseph, MI: ASAE.
- Fedrizzi, M., Fedrizzi, M. & Ostasiewicz, W. (1993). Towards fuzzy modeling in economics. *Fuzzy Sets and Systems* 54(3): 259–268.
- Flanagan, M. (1989). On the application of approximate reasoning to control of the activated sludge process. *Proceedings of the Joint Automatic Control Conference*, Vol. 31, pp. 13–22.

- Gardin, F., Power, R. & Martinelli, E. (1995). Liquidity management with fuzzy qualitative constraints. *Decision Support Systems* **15**(2): 147–156.
- Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Menlo Park, CA: Addison-Wesley.
- Gustafson, D. & Kessel, W. (1984). Fuzzy clustering with a fuzzy covariance matrix. *Proceedings IEEE CDC*, Vol. 12, pp. 1–25.
- Hwang, H. & Woo, K. (1995). Linguistic fuzzy model identification. *IEEE Proceedings-Control Theory and Applications* **142**(6): 537–544.
- Jang, J. (1993). ANFIS – adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics* **23**(3): 665–685.
- Jang, J. & Sun, C. (1995). Neuro-fuzzy modeling and control. *Proceedings of the IEEE* **83**(3): 378–406.
- Karr, C. (1991a). Applying genetics to fuzzy logic. *AI Expert* **March**: 38–43.
- Karr, C. (1991b). Genetic algorithms for fuzzy controllers. *AI Expert* **February**: 26–33.
- Karr, C. & Gentry, E. (1993). Fuzzy control of ph using genetic algorithms. *IEEE Transactions on Fuzzy Systems* **1**(1): 46–53.
- Keller, J. & Hunt, D. (1985). Incorporating fuzzy membership functions into the perception algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-7**(6): 693–699.
- Keller, J. & Tahani, H. (1992). Implementation of conjunctive and disjunctive fuzzy logic rules with neural networks. *International Journal of Approximate Reasoning* **6**: 221–240.
- Konstantinov, K. & Yoshida, T. (1989). Physiological state control of fermentation processes. *Biotechnology and Bioengineering* **33**: 1145–1156.
- Konstantinov, K. & Yoshida, T. (1990a). An expert approach for control of fermentation processes as variable structure plants. *Journal of Fermentation and Bioengineering* **70**(1): 48–57.
- Konstantinov, K. & Yoshida, T. (1990b). On-line monitoring of representative structural variables in fed-batch cultivation of recombinant *escherichia coli* for phenylalanine production. *Journal of Fermentation and Bioengineering* **70**(6): 420–426.
- Konstantinov, K. & Yoshida, T. (1992). Knowledge-based control of fermentation processes. *Biotechnology and Bioengineering* **39**: 479–486.
- Kosko, B. (1992). *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs, NJ: Prentice Hall.
- Kosko, B. (1993). *Fuzzy Thinking: the New Science of Fuzzy Logic*. New York: Hyperion.
- Kruse, R., Gebhardt, J. & Klawonn, F. (1994). *Foundations of Fuzzy Systems*. Chichester: John Wiley & Sons, Inc..
- Lasek, M. (1992). Hierarchical structures of fuzzy ratings in the analysis of strategic goals of enterprises. *Fuzzy Sets and Systems* **50**(2): 127–134.
- Lee, C. (1990). Fuzzy logic in control systems: Fuzzy logic controller-part i. *IEEE Transactions on Systems, Man, and Cybernetics* **20**(2): 404–417.
- Linko, P. (1988). Uncertainties, fuzzy reasoning, and expert systems in bioengineering. *Annals of the New York Academy of Sciences* **542**: 83–101.
- Mamdani, E. & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man Machine Studies* **7**(1): 1–13.
- Nakamishi, S., Takagi, T. & Kurosania, M. (1993). Expert systems of beef grading by fuzzy inference and neural networks. *Proceedings of the 3rd International Fuzzy Association World Congress*, pp. 360–375.
- Norwich, A. & Turksen, I. (1984). A model for the measurement of membership and the consequences of its empirical implementation. *Fuzzy Sets and Systems* **12**: 1–25.
- Ntuen, C. & Chestnut, J. (1995). An expert-system for selecting manufacturing workers for training. *Expert Systems with Applications* **9**(3): 309–332.
- Park, D., Kandel, A. & Langholz, G. (1994). Genetic-based new fuzzy reasoning models with application to fuzzy control. *IEEE Transactions on Systems, Man, and Cybernetics* **24**(1): 39–47.

- Procyk, T. & Mamdani, E. (1979). A linguistic self-organizing process controller. *Automatica* **15**(1): 15–30.
- Qian, Y. & Tessier, P. (1995). Application of fuzzy relational modeling to industrial-product quality-control. *Chemical Engineering & Technology* **18**(5): 330–336.
- Schmoldt, D. (1991). Simulation of plant physiological processes using fuzzy variables. *AI Applications* **5**(4): 3–16.
- Sugeno, M. & Yasukawa, T. (1993). A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems* **1**(1): 7–31.
- Takagi, H. & Hayashi, I. (1991). NN-driven fuzzy reasoning. *International Journal of Approximate Reasoning* **5**(3): 191–212.
- Tansheit, R. & Scharf, E. (1988). Experiments with the use of a rule-based self-organizing controller for robotics applications. *Fuzzy Sets and Systems* **26**: 195–216.
- Thangaradivehu, S. and Colvin, T. (1993). Trafficability determination using fuzzy set theory. *Transactions of the ASAE* **34**(5): 2272–2278.
- Tog (1993). *TILShell User Manual*, 3.0.0 edn.
- Turksen, I. (1991). Measurement of membership functions and their acquisition. *Fuzzy Sets and Systems* **40**: 5–38.
- Turksen, I., Tanaka, H. & Watada, J. (1993). Special issue on expert decision-support systems – foreword. *Fuzzy Sets and Systems* **58**(1): 1–2.
- Valenzuela, M. (1992). The fuzzy classifier system: Motivations and first results. *Parallel Processing from Nature* **1**(1): 338–342.
- Verma, B. (1995). Application of fuzzy logic in postharvest quality decisions. *Proceedings of the National Seminar on Postharvest Technology of Fruits*. Bangalore, India: University of Agricultural Sciences.
- Wiggins, R. (1992). Fuzzy control: A genetic approach. *AI Expert* **May**: 28–35.
- Williams, T. (1994). New tools make fuzzy/neural more than an academic amusement. *Computer Design* **July 14**: 69–84.
- Yager, R. & Filev, D. (1994). *Essentials of Fuzzy Modeling and Control*. New York: John Wiley & Sons, Inc.
- Yager, R., Goldstein, L. & Mendels, E. (1994). Fuzmar – an approach to aggregating market-research data-based on fuzzy-reasoning. *Fuzzy Sets and Systems* **68**(1): 1–11.
- Yamakawa, T. (1992). A fuzzy logic controller. *Journal of Biotechnology* **24**: 1–32.
- Yang, M. (1993). A survey of fuzzy clustering. *Mathematical and Computer Modelling* **18**(11): 1–16.
- Yoshinari, Y., Pedrycz, W. & Hirota, K. (1996). Construction of fuzzy models through clustering-techniques. *Fuzzy Sets and Systems* **78**(1): 1–4.
- Zadeh, L. (1965). Fuzzy sets. *Information and Control* **8**(1): 338–353.
- Zadeh, L. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-3**: 28–44.
- Zadeh, L. (1975a). The concept of a linguistic variable and its application to approximate reasoning: part 1. *Information Sciences* **8**(1): 199–249.
- Zadeh, L. (1975b). The concept of a linguistic variable and its application to approximate reasoning: part 2. *Information Sciences* **8**(1): 301–357.
- Zadeh, L. (1976). The concept of a linguistic variable and its application to approximate reasoning: part 3. *Information Sciences* **9**(1): 43–80.

Robotics for Plant Production

NAOSHI KONDO¹ and K.C. TING^{2,*}

¹*Agricultural Engineering Dept., Faculty of Agriculture, Okayama University, 1-1-1, Tsushima-Naka, Okayama, Japan, E-mail: nkondo@ccews2.cc.okayama-u.ac.jp;*

²*Dept. of Bioresource Engineering, Rutgers University-Cook College, P.O.Box 231, New Brunswick, New Jersey 08903-0231, USA, E-mail: ting@bioresource.rutgers.edu, Tel: (732) 932-9753, Fax: (732) 932-7931 (* Author for correspondance)*

Abstract. Applying robotics in plant production requires the integration of robot capabilities, plant culture, and the work environment. Commercial plant production requires certain cultural practices to be performed on the plants under certain environmental conditions. Some of the environmental conditions are mostly natural and some are modified or controlled. In many cases, the required cultural practices dictate the layout and materials flow of the production system. Both the cultural and environmental factors significantly affect when, where and how the plants are manipulated. Several cultural practices are commonly known in the plant production industry. The ones which have been the subject of robotics research include division and transfer of plant materials in micropropagation, transplanting of seedlings, sticking of cuttings, grafting, pruning, and harvesting of fruit and vegetables. The plants are expected to change their shape and size during growth and development. Robotics technology includes many sub-topics including the manipulator mechanism and its control, end-effector design, sensing techniques, mobility, and workcell development. The robots which are to be used for performing plant cultural tasks must recognize and understand the physical properties of each unique object and must be able to work under various environmental conditions in fields or controlled environments. This article will present some considerations and examples of robotics development for plant production followed by a description of the key components of plant production robots. A case study on developing a harvesting robot for an up-side-down single truss tomato production system will also be described.

Key words: automation, end-effector, manipulator, plant cultural practice, robotics, traveling device, visual sensor

Introduction

Robots have been playing an important role in automated materials handling in the manufacturing industry. Service robots have also gained attention in recent years. For agricultural production, robotics has been an active research topic in several parts of the world for more than 15 years. A substantial amount of effort in agricultural robotics development is aimed at improving automation in operations related to the production of plants. Robotics technology includes many sub-topics including the manipulator mechanism and its control, end-

effector design, sensing techniques, mobility, and workcell development. Economic issues and the man-machine interface are also important.

Plant production occurs at many levels, ranging from micropropagation for plant regeneration to large-scale production of field crops and forest trees. In current commercial plant production systems, plant materials are frequently handled in large quantities. The need to use machinery in plant production to reduce human labor requirements, expand production capabilities, and improve uniformity of products has long been recognized. The mechanization of plant production enables a small percentage of the population who are modern farmers to produce large quantity of food, fiber, and ornamentals on a desired schedule. Owing to the biological characteristics of plants, they need to be manipulated in special ways, and sometimes with individual care. The cost of materials handling by manual labor and the availability of skilled personnel have become an increasing concern for plant producers. The advent of computers brought about the possibility of equipping machines with a certain degree of intelligence. A robot is a programmable intelligent machine equipped with sensing devices and changeable end-effectors, positioned and oriented by a mechanism, for performing multiple materials handling tasks. In addition to mechanization functions, robots also provide various automation capabilities with flexibility, which can be of great value to plant production operations.

In a broad sense, automation encompasses machine capabilities of information processing and task execution to facilitate a system's operation. Information processing includes the activities of information acquisition, organization, manipulation, interpretation, understanding, adoption, and presentation. Commonly applied information processing functions in an automated system are perception, reasoning, learning, and communication (Ting and Giacomelli 1992; Miles 1994). Task execution requires task planning and mechanical work. As mentioned above, robotics is an integration of computer technologies, sensing and control techniques, and generic mechanisms. The result is a flexibly automated mechatronic system well suited for performing, with some intelligence, various materials handling operations. Robots are normally working in concert with other sensing and materials handling devices within a defined space, e.g. workcell. The types of devices included in a workcell determines its overall functionality. Therefore, it is advisable to give careful considerations at the systems level before the actual implementation of the workcell design.

Applying robotics in plant production requires the integration of robot capabilities, plant culture, and the work environment. This article will present some considerations and examples of robotics development for plant production followed by a description of the key components of plant production

robots: manipulator, end-effector, visual sensor, and travelling device. A case study on developing a harvesting robot for an up-side-down single truss tomato production system will also be described. Since most of the work on robotics for plant production to date is for horticultural crops, the emphasis of this article will therefore be on horticultural robots.

Horticultural Robots

Robots in manufacturing industries have become popular and have been studied extensively. Most used in factories usually consist of a manipulator and end-effector and can work under feedback or sequential control. Their work objects have certain, generally well defined, sizes, shapes, color, hardness, and texture. The work environments for industrial robots are mostly well structured regarding the lighting, layout, and materials handling procedures.

Commercial plant production requires certain cultural practices to be performed on the plants under certain environmental conditions. Some of the environmental conditions are mostly natural and some are modified or controlled. In many cases, the required cultural practices dictate the layout and materials flow of the production system. Both the cultural and environmental factors significantly affect when, where and how the plants are manipulated. In addition, the plants are expected to change their shape and size during growth and development. Individual plants within any given population will have significant variation in properties important to the robotic operation to be performed. The robots which are to be used for performing plant cultural tasks must recognize and understand the physical properties of each unique object and must be able to work under various environmental conditions in fields or controlled environments. They often need, therefore, sensing systems which can work under the variable conditions as well as specialized manipulators and end-effectors. The environmental conditions are occasionally so severe with regard to high temperature, humidity, dust and/or rain that electrical circuit and material corrosion problems can be major concerns. These conditions must be taken into consideration when designing or selecting plant production robot systems. In addition, when the work object is not easily positioned in front of the robot, a travelling device is required.

Several cultural practices are commonly known in the plant production industry. The ones which have been the subject of robotics research include division and transfer of plant materials in micropropagation, transplanting of seedlings, sticking of cuttings, grafting, pruning, and harvesting of fruit and vegetables.

Kozai et al. (1991) described the special characteristics of micropropagation procedures and the considerations for automation. A number of robotic

based systems for multiplication of plant tissues and transplanting of plantlets developed by commercial companies and research institutions have been introduced. Okamoto et al. (1992) developed a tissue proliferation robot for dividing and transferring callus. A special end-effector equipped with a disinfected pair of razor blades and a suction pipe was carried by an articulated robot to perform the task. The robot was guided by a machine vision system in locating plant tissues on a petri dish. The average cycle time for dividing and transfer a callus was approximately 40 seconds.

There have been a series of projects in developing robotic systems for transplanting seedling plugs and cuttings. Transplants are used in both greenhouse and field production of floral and vegetable crops because of their many advantages including uniformity, earliness, etc. The additional labor required in handling transplants, as opposed to direct seeding, is the transplanting operation. Kutz et al. (1987), Ting et al. (1990), Yang et al. (1991), Tai et al. (1994), and Bar et al. (1996) have done work in using robots for transplanting plugs. Simonton (1990) developed a robotic workcell for geranium stock processing. Geranium cuttings were processed using this system for vegetative propagation. A grafting robot was developed by Suzuki et al. (1993) for preparing cucumber seedlings. The grafting operation involved the preparation of scions and fixing/adhering the scions on stocks. It was reported that the cycle time for producing a grafted seedling was about 3 seconds. Yamada et al. (1995) also developed a fully automated robot-based grafting robot.

Sevila (1985) studied an alternative grape vine pruning method to facilitate robotic applications. The new pruning method was evaluated and revised by a computer model and a laboratory scale robotic system was constructed. The system consisted of a cutting saw attached to a cutting arm, an image acquisition system, and an electronic controller. It was concluded that the method was physiologically and agronomically feasible. The robotic system was found workable with the new pruning method. At Cornell University, researchers have worked on the development of a robotic grape pruner. A computer model of a vision guided pruner was developed by Ochs and Gunkel to study the effects of the robot components and the variation of vine and terrain on the pruning accuracy (Ochs and Gunkel 1993). The design of a digital regulator and tracking controller for a robotic electro-hydraulic pruner was discussed by Lee et al. (1994). In related research, work has been done at Okayama University to develop a robot which would perform a number of operations in vineyard (Monta et al. 1994). The operations that the robot was capable of doing included spraying, bagging, berry thinning, and harvesting.

Harvesting of fruit and vegetables using robots has been a popular subject of research and development. Research works have been reported since

1983 on harvesting a variety of crops using robotic systems. The types of fruit and vegetables harvested included apple (d'Esnon 1985; Bourely et al, 1991), orange (Coppock 1983; Slaughter and Harrell 1987), peach (Vassura 1991), melon (Edan et al. 1994), watermelon (Tokuda et al. 1995; Iida et al. 1995), cucumber (Arima et al. 1995), tomato (Monta et al. 1996), cabbage (Murakami et al. 1995), and mushroom (Reed et al. 1995). Kondo et al. (1994) developed several robotic harvesting hands for fruit vegetables including tomatoes, cherry tomatoes, and cucumbers.

Key Robotic Components

Manipulator

The basic mechanism of a manipulator is defined by its degrees of freedom, the type of joint, link length, and offset length. Any kind of manipulator may be used, if its work envelope includes the position of the work object and the work efficiency is not a major concern. This is because the principle task of a manipulator is to move and orient an end-effector to a position where it can interact with the work object. If a manipulator whose basic mechanism is not optimized for a particular production system is used, the work speed may be slow and the manipulator may only be able to place the end-effector at a singular point. This can potentially present some problems in a plant production system. The manipulator may have to risk a collision with the objects within the work envelope. Therefore, a mechanism optimized for the specific task is often required for a plant production robot. On the other hand, however, a manipulator which has a mechanism developed based on a specific operation is likely to have less flexibility in adapting to other operations. Nevertheless, a special purpose manipulator can still be used in performing various jobs by using different end-effectors. The factors normally considered in determining basic mechanism requirements for a manipulator of a plant production robot will be described in the following.

Work envelope:

As mentioned above, the role of a manipulator is to send an end-effector to a 3-D position within its work envelope. It is very common to have work objects presented by some sort of conveying system to a robot on a 2-D plane within a workcell. Work objects such as plants with fruit to be harvested or trays of plants to be worked on can be presented to a robot in the similar way. In such a case, many types of manipulator mechanism may be used; however, an appropriate spatial range for the positions of the work objects should be included within the work envelope. For field production, a large

work envelope is normally used to reduce the need of moving the manipulator. However, the most appropriate link length and degrees of freedom should be considered according to the application. In this case, the following normalized volume index V_n is used:

$$V_n = V/(4\pi L^3/3) \quad (1)$$

where V is the volume of the work envelope and L is the total link length. It is, however, better to consider not only this index but the shape of the work envelope adaptable to the expected range of positions of work objects. This shape depends on the type and specific design of the manipulator.

Measure of manipulatability:

The configuration of a manipulator is evaluated by the measure of manipulatability which implies easiness to move the end of the manipulator. The measure of manipulatability, w , is defined, based on the Jacobian of the manipulator joint variables $J(\Theta)$, by Equation (2) (Yoshikawa 1983):

$$w = (\det(J(\Theta)J^T(\Theta)))^{1/2} \quad (2)$$

When a plant has many fruits to be harvested, this index is important for a high work efficiency of the harvesting robot. For example, when the elbow angle is 90 degrees in case of a 2 DOF manipulator consisting of elbow and wrist joints, the measure of manipulatability is maximum. It is understandable that a human's hand is easily moved when its elbow is bent at around 90 degrees. The basic mechanism should be determined so that the measure of manipulatability has a large value when the manipulator takes a configuration for operation.

Posture diversity:

If a manipulator has more than 7 DOF, it has redundant space. This means that the manipulator can have an access to the object through plural routes. It also can be said that the manipulator can have the possibility to avoid an obstacle before approaching the target object if necessary. The posture diversity is defined by the angle at the redundant space as shown in Figure 1 (Okamoto et al. 1992; Kondo et al. 1993).

There are several other evaluation indices for the mechanism of a manipulator such as: positioning accuracy of the manipulator end, ease of control of the manipulator and so on (Okamoto et al. 1992). Figure 2 shows a mechanism for a tomato harvesting manipulator (Kondo et al. 1993) as an example using the above evaluation indices. This is based on the conventional tomato plant training system like the configuration of a fence on a vertical plane in which tomato plants are grown until six or seven trusses of fruit are harvested.

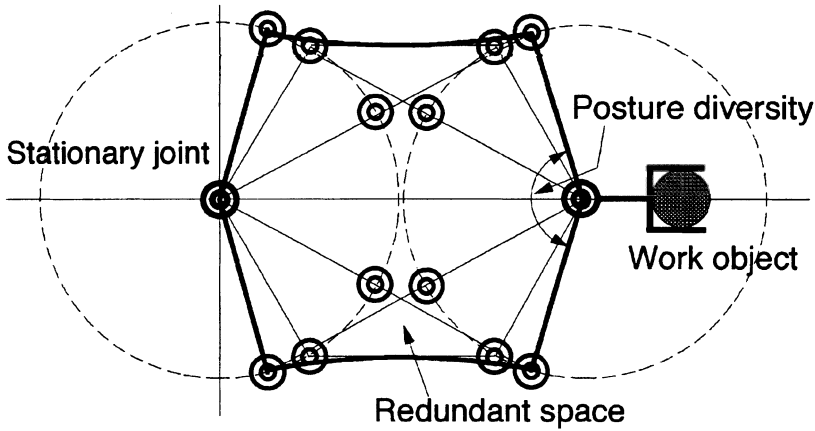


Figure 1. Posture diversity.

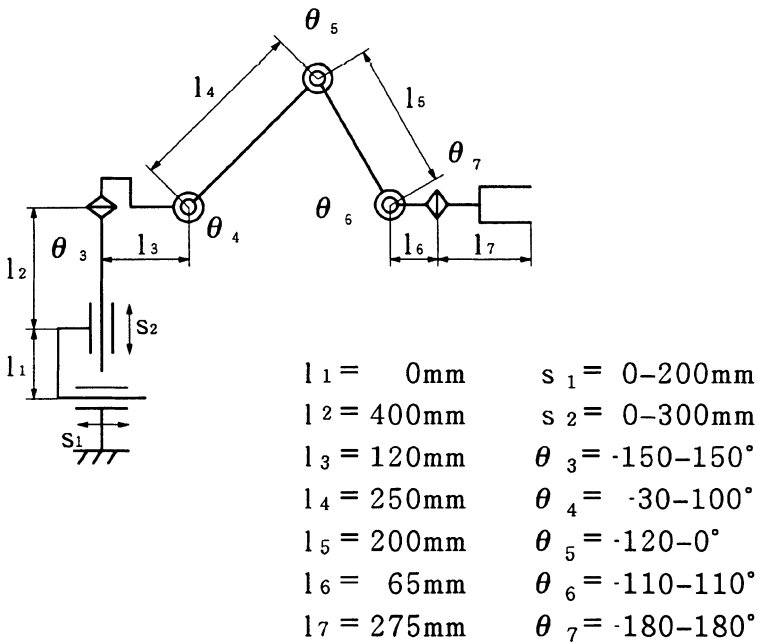


Figure 2. Basic mechanism of a tomato harvesting manipulator.

This manipulator shown has 7 degrees of freedom. If this manipulator is used for a harvesting operation it is easy for the manipulator end to approach any fruit in the plant with high manipulatability.

End-effector

A specific mechanism for an end-effector depends on a specific work object and an operation to be performed. Since the properties of the work object the robot directly handles and the type of operation are different from any others this should be unique. Most of objects for plant production systems have various sizes and uncertain shapes, at least to some degree, even if they are same varieties. They are usually much softer than the materials of the robot and are usually easily damaged. Before designing a specific end-effector, therefore, many kinds of physical properties of its intended work object should be measured such as: shape, size, mass, cutting resistance, frictional resistance, elasticity, viscosity and so on according to the operation to be performed. When those properties are not the only ones required to develop the end-effector, optical, sonic and electrical properties are also often required. In addition, chemical and biological properties sometimes may be necessary. The end-effector developed based on such specific properties is usually not for multi-purpose use but for specific use to achieve highly efficient work. It often has to have sensors so as not to injure the object and to compensate for errors of another sensor, since handling of the object by the end-effector influences product market value.

Figure 3 shows a cherry tomato harvesting end-effector (Kondo et al. 1995). This end-effector pulls a fruit into its tube opening pneumatically by suction. Three pairs of photo-sensors detect the fruit location in the end-effector. If the fruit comes to an appropriate position, its peduncle will be nipped at the joint by the action of the nipper closing. The fruit which has been detached from its peduncle will be transported to a container through the tube by suction. Small fruit like cherry tomatos or strawberries can be harvested by this kind of end-effector. However, the tube must be designed so that fruit is not injured when it is transported.

Visual sensor

A visual sensor is a very important external sensor of a robot just like the eyes for humans. The three important functions of visual sensors for a plant production robot are: discrimination, recognition, and distance measurement. The work objects may have specific optical properties (as the examples shown in Figure 4), uncertain shape, and various positioning possibilities. Figure 4 shows that there are differences in the color of plant material within the visible region which a human being can normally detect. Plant reflectance in the infrared region depends on the part of plant from which energy is reflected. It is interesting that reflectances of fruits are classified into two groups. One group has higher reflectance than that of leaves within the waveband of

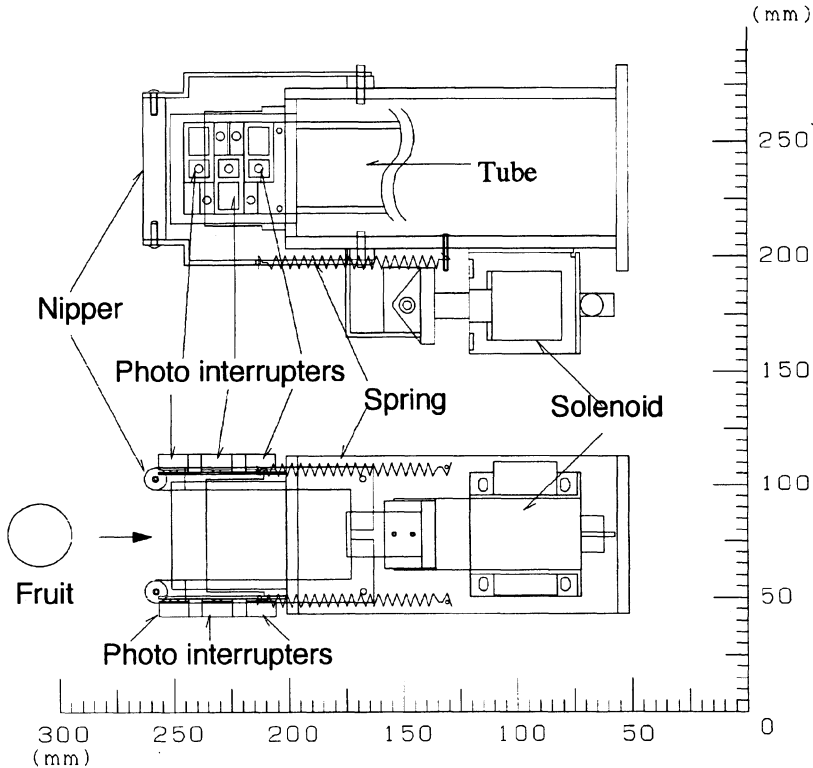


Figure 3. End-effector for cherry tomato harvesting.

700 nm to 1100 nm. The reflectance of the other group is lower than that of leaves. The difference seems to be caused by the difference of the state of water in the surface of fruit, since most of fruits are juicy in the latter group. Fruit and stem have water absorption bands at 970 nm and 1170 nm in their reflectances. Flowers and leaves have no absorption band because of their thinness. Most plants have these characteristics of reflectances as shown in Figure 4, so these characteristics should be used for effective discrimination of the work object by a visual sensor.

Discrimination:

When the color of a work object is different from that of the others, it is not so difficult to discriminate the work object in the image by using R, G, B signals from a color TV camera. For example, ripe tomato fruit and leaves can be discriminated by comparing red with green signals. When unripe tomato fruit is discriminated from green color leaves and stems, it is necessary to

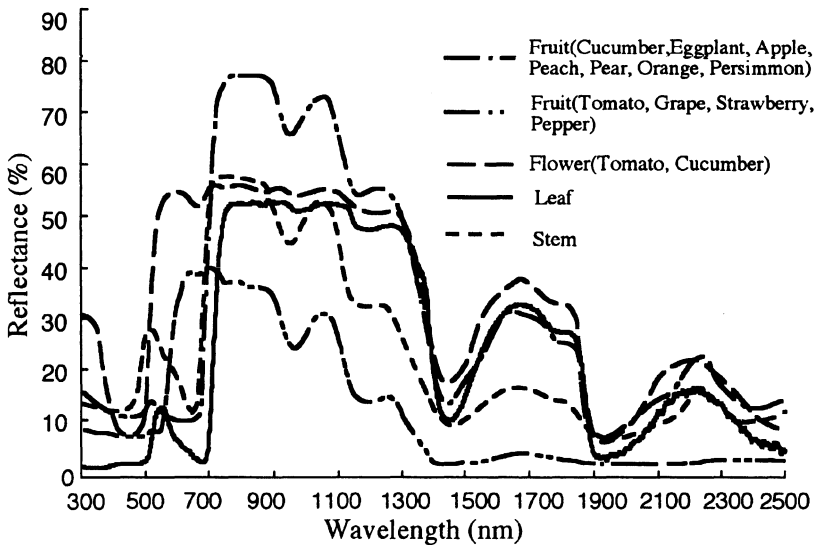
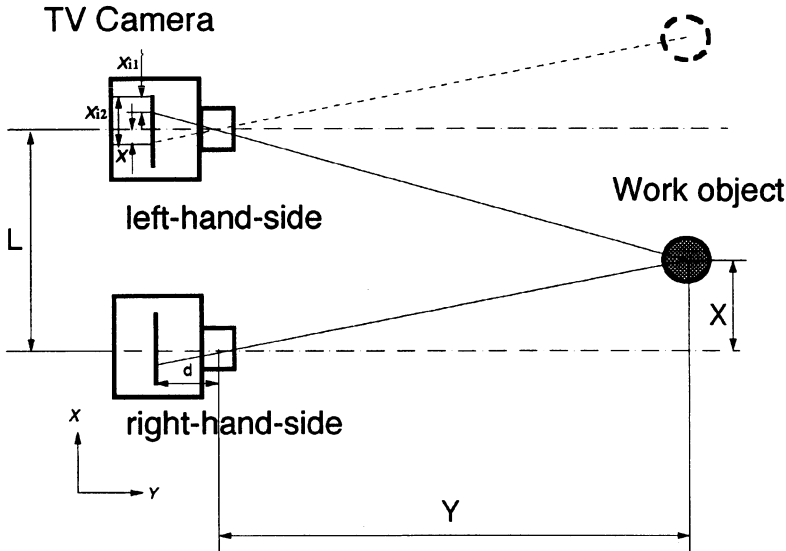


Figure 4. Spectral reflectance of plant materials.

acquire images through 670 nm and 970 nm interference filters (Kondo and Endo 1987). The 550 nm and 850 nm interference filters are effective for discrimination of green cucumber fruit (Kondo and Endo 1987; Kondo et al. 1994), where 670 nm is the chlorophyll absorption band, 970 nm is the water absorption band, 550 nm is the center of wavelength for the green color and 850 nm is the wavelength at which there is significant difference in reflectance between fruit and leaf. These wavelengths are so important that even green color fruit can be effectively discriminated from the others in the acquired image.

Recognition:

A thresholded image which mainly consists of work objects can be obtained after comparing the images acquired through the same specific optical filters as used for discrimination. It is necessary to recognize the size, direction, shape, number and so on of work objects in the image for handling the object. Before recognizing the blob, or outline of the object, some processing is sometimes conducted for the binary image such as smoothing, contraction, dilatation, thinning, border following, edge detection, noise reduction and so on. This is because the image often has not only the work object but also noise and unnecessary substances. When the characteristics of the object are recognized, it is important to extract some features from the image. The features for shape recognition are very many, such as area, perimeter, Feret's diameters, moment,



Z direction is perpendicular to the x - y plane.

Figure 5. Binocular stereo vision.

fractal dimension, intersection, orientation and so on. Furthermore, textual features extracted from gray level image are sometimes necessary such as angular second moment, contrast, inverse difference moment, correlation and so on (Haralick et al. 1973).

Distance measurement:

Distance measurement is essential for robotic operation. Several methods based on the principle of triangulation have been reported previously. The most well-known method is binocular stereo-vision. If two images (of the same object) are acquired at different places, the distance from the visual sensor to objects can be measured. The distances Y and X in Figure 5 are calculated by Equation (3).

$$Y = dL/(x_{i2} - x_{i1}), \quad X = xY/d \quad (3)$$

A visual sensor can be attached to the manipulator end of a robot system, because the position of a visual sensor can be strategically changed to utilize the pixel number of visual sensor efficiently and to have possibility to detect the object hidden by an obstacle. When the visual sensor is attached

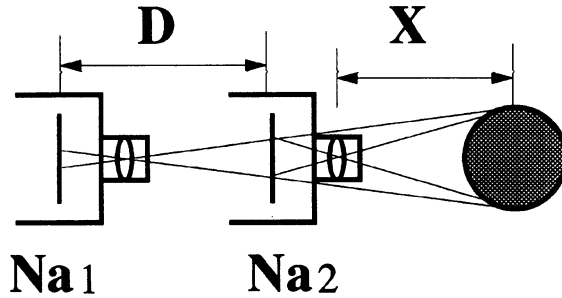


Figure 6. Differential object size method.

to the manipulator end, the sensor moves to the object with the manipulator approaching. The distance X in Figure 6 is calculated by Equation (4), because the number of pixels representing the object is increasing when the manipulator is moving toward the object.

$$X = D(Na1)^{1/2} / ((Na2)^{1/2} - (Na1)^{1/2}) \quad (4)$$

where Na_i is the number of pixels representing the object in the i -th image ($i = 1, 2$), and D is the visual sensor moving distance.

These methods are classified into a category of passive range finder method. On the other hand, there is an active range finder method measuring by projecting light and receiving it. If the light is scanned horizontally and vertically, a 3-D sensing capability can be achieved. In addition, if a red light beam and infrared light beam are used as the projected lights, not only distance information but color information can be obtained (Fujiura et al. 1992).

One more important thing is that the visual sensor for a plant production robot is often used in the field or in a greenhouse where illuminance and the color temperature of sunlight are changing from time to time. The influence of the various conditions of sunlight should be eliminated. Using a pair of optical filters based on the spectral reflectance of an object can solve these problems. This is specially important in a greenhouse where the visual sensor may be used under the condition of high temperature and high humidity. Needless to say, the most simple, light and inexpensive sensor is desired.

Travelling device

When the work object is a small and portable object, such as a harvested fruit, a seedling or its tray, the robot can be stationary (Ting et al. 1990). In such circumstances the work object is much more easily transported than the

robot. The robot itself can also be stationary in the cases where the work object is positioned for the robot independantly, as with a conveyor or some other simple, non-robotic positioning device. However, the robot often has to move by itself in the field or from place to place within a greenhouse. This is required when the object is not portable, such as fruit, flower or grain attached to a plant which is itself immovable in the soil, or when the operation should be conducted in the field. In such a case, the robot needs its own travelling device.

Automatic controls of the most common types of traveling devices of wheel (Yamashita et al. 1992), rail (Itokawa 1990), crawler (Kondo 1993) and gantry system (Miyazawa 1987) have been reported. The wheel type has a simple structure and is easily used, so some autonomous vehicles for applications such as sprayer operation or simple transport have already been commercialized. The rail type traveling device is effective in the intensive production areas in structures such as greenhouses or terraced orchards. The initial cost for the rail is quite expensive and maintenance of the rail is necessary, but such a device is relatively easy to control. The crawler type is needed for the transportation of large robots to reduce pressure on ground. In some fields where a gantry system can be installed, it is relatively easy to introduce a robot and it is not necessary to worry about pressure on the ground, because its wheels and rails are on levees of the field. In a gantry system, high precision for positioning and stability of robotic motion can be obtained, however there are fields where the gantry system installation is difficult.

Horticultural Approach for Robotic Development

As described above, each component for a plant production robot must be developed based on the physical properties of its work object. The horticultural aspects of the production system must also be considered to realize the practical use of various kinds of robots. For example, plant training systems and cultivation methods have been developed so that productivity and quality can be improved and that the farmer can work easily. But the robot cannot often work efficiently in the present training system, since its eye, arm, hand and leg are inferior to those of the farmer. These factors can prevent the practical use of the robot. There are, therefore, trials underway to change the conventional plant training system into a new training system adaptable not only to farmer's operation but also to robotic operation. Cultural systems can be developed to facilitate cooperative work between human operators and robots.

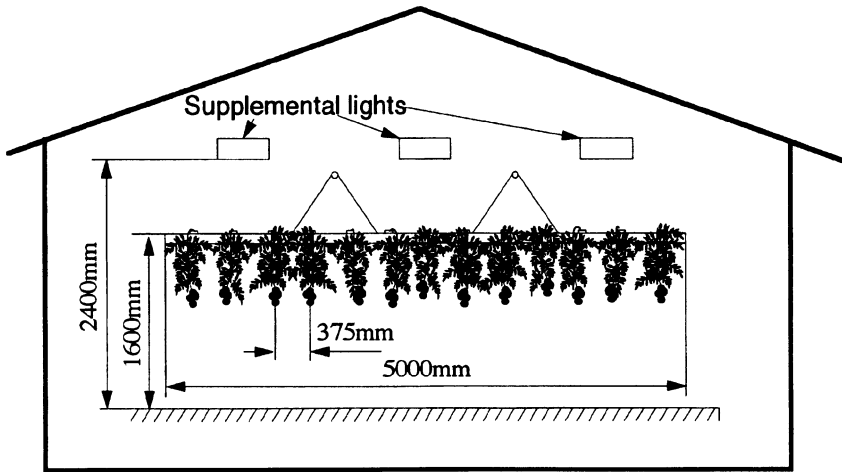


Figure 7. Upside down single truss tomato production system.

Figure 7 shows a new plant training system for tomato plants (Monta et al. 1996). This is called the upside down single truss tomato production system. The plant and fruit trusses hang down from the trough so that robot can find fruits and can have an access to fruits easily from underneath or the sides. This system was proposed to realize several potential advantages as follows: (1) tomato plants including seedling, truss, fruit, leaf and stem can be standardized, (2) uniform quality and quantity are expected, (3) production scheduling is easy and demand of labor is predictable, (4) a transportable bench or flume is easily used and greenhouse space is efficiently utilized, (5) mechanization is relatively easy, and (6) much less labor for plant training is required.

Configuration of some plants can be also changed during growing under controlled environment. When difference of temperature between day time and night time is appropriately controlled (Heins and Erwin 1990) or when irrigation or fertilization is changed, morphological features of the plants are changed. Appropriate environmental control for a plant production robot may be required for its practical application. From another viewpoint, such control can facilitate standardization of plants. It is expected that the robot can work much more easily, if the size, shape, position and growth habit of the objects have been standardized by environmental control. In addition, if a data base for a standardized plant can be accumulated as knowledge for the robot, the possibilities for intelligence based control algorithms for the robot increase.

It will also be important to choose a suitable variety for robotic operation. For example, an experimental result of robot harvesting says that a longer

peduncle is better for robot harvesting of many fruits (Kondo et al. 1994). If the peduncle is too short, there is a risk to injure fruits or stems and the success rate of harvesting may decrease. A new variety of plant may be developed for robotic operation based on the genetic engineering approach in the near future.

There have been many trials to change training systems and morphological features for plant production robots for harvesting crops such as tomato, cherry tomato, cucumber, strawberry, etc. When a plant production robot is to be developed, consideration of effectively integrating horticultural technologies and robotic technologies is essential.

Conclusion

The plant production robots are different from the industrial robots mainly due to the need to integrate engineering technologies with agricultural/horticultural technologies. Both robotics and horticultural technologies are now progressing rapidly. It is, therefore, important that engineers and plant scientists work closely together in the development of robotics for plant production. Furthermore, economic consideration is also an important aspect of robotic development. There have been many research and development projects on horticultural robotics with impressive results; however, the utilization of robotics in plant production is still limited compared to its use in the manufacturing industry. There are undoubtedly some interesting challenges and exciting opportunities existing in the development of robotics for plant production.

Acknowledgment

New Jersey Agricultural Experiment Station Publication No. D-03232-40-96.

References

- Arima, S., Kondo, N., Fujiura, T., Nakamura, H. & Yamashita, J. (1995). Basic studies on cucumber harvesting robot. *Proceedings of ARBIP95* vol. 1, 195–202. Japan Society of Agricultural Machinery.
- Bar, A., Edan, Y. & Alper, Y. (1996). Robotic transplanting: simulation and adaptation. Paper no. 963008. St. Joseph, MI: ASAE.
- Bourelly, A., Rabatel, G., & Sevilla, F. (1991). A mobile robot with two actuators to pick apples. *The Second Workshop on Robotics in Agriculture & the Food Industry*, 229–237. Italy: DIST University of Genova.
- Coppock, G. E. (1983). Robotic principles in the selective harvesting of valencia oranges. *Robotics and Intelligent Machine in Agriculture*, 138–145. St. Joseph, MI: ASAE.

- d'Esnon, A. G. (1985). Robotic harvesting of apples. *Agri-Mation 1*, 210–214. St. Joseph, MI: ASAE.
- Edan, Y., Wolf, I., Grinshpun, J., Dobrusin, Y., & Rogozin, V. (1994). Robotic melon harvesting: prototype and field tests. Paper no. 943073. St. Joseph, MI: ASAE.
- Fujiura, T., Yamashita, J., & Kondo, N. (1992). Agricultural robots(1)-vision sensing system. Paper No. 923517. St. Joseph, MI: ASAE.
- Haralick, R. M., Shanmugam, K. & Dinstein, I. (1973). Textural features for image classification. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-3(6): 610–621.
- Heins, R. & Erwin, J. (1990). Understanding & applying DIF. *Greenhouse Grower*, 73–78. February.
- Iida, M., Namikawa, K., Furube, K., Umeda, M., & Tokuda, M. (1995). Development of watermelon harvesting robot (II)-watermelon harvesting gripper. *Proceedings of ARBIP95* vol. 2, 17–24. Japan Society of Agricultural Machinery.
- Itokawa, N. (1990). Development of spray device on mono-rail. *Journal of Japanese Society of Agricultural Machinery (Kansai-branch)* 67: 25–28.
- Kondo, N. (1993). Basic studies on robot to work in vineyard (Part 1). *Journal of the Japanese Society of Agricultural Machinery* 55(6), 85–94.
- Kondo, N. & Endo, S. (1987). Visual sensor for recognizing fruit (Part 1). *Journal of the Japanese Society of Agricultural Machinery* 49(6): 563–570.
- Kondo, N., Monta, M., Shibano, Y. & Mohri, K. (1993). Basic mechanism of robot adapted to physical properties of tomato plant. *Proceedings of International Conference for Agricultural Machinery and Process Engineering* 3: 840–849.
- Kondo, N., Monta, M., Shibano, Y., Mohri, K. & Arima, S. (1994). Robotic harvesting hands for fruit vegetables. Paper no. 943071. St. Joseph, MI: ASAE.
- Kondo, N., Nakamura, H., Monta, M., Shibano, Y. & Mohri, K. (1994). Visual Sensor for Cucumber Harvesting Robot. *Proceedings of Processing Automation Conference III*, 461–470.
- Kondo, N., Fujiura, T., Monta, M., Shibano, Y., Mohri, K. & Yamada, H. (1995). End-effectors for petty-tomato harvesting robot. *Acta Horticulturae* 399: 239–245.
- Kozai, T., Ting, K. C. & Aitken-Christie, J. (1991). Considerations for automation of micro-propagation systems. *Automated Agriculture for the 21st Century*, 503–517. St. Joseph, MI: ASAE.
- Kutz, L. J., Miles, G. E., Hammer, P. A. & Krutz, G. W. (1987). Robotic transplanting of bedding plants. *Transactions of the ASAE* 30(3): 586–590.
- Lee, M. F., Gunkel, W. W. & Throop, J. A. (1994). A digital regulator and tracking controller design for a electro-hydraulic robotic grape pruner. *Computers in Agriculture-Proceedings of the 5th International Conference*, 23–28. St. Joseph, MI: ASAE.
- Miles, G. E. (1994). Automation basics: perception, reasoning, communication, planning, and implementation. *Greenhouse Systems-Automation, Culture, and Environment*, 8–15. Ithaca, NY: NRAES-72, Northeast Regional Agricultural Engineering Service.
- Miyazawa, F. (1987). Gantry system. *Proceedings of International Symposium on Agricultural Mechanization and International Cooperation in High Technology Era*, 109–114. Japanese Society of Agricultural machinery.
- Monta, M., Kondo, N., Shibano, Y. & Mohri, K. (1994). Study on a robot to work in vineyard. Paper no. 943072. St. Joseph, MI: ASAE.
- Monta, M., Kondo, N., Ting, K. C., Giacomelli, G. A., Mears, D. R. & Kim, Y. (1996). End-effector for tomato harvesting robot. Paper no. 963007. St. Joseph, MI: ASAE.
- Murakami, N., Inoue, K. & Ootsuka, K. (1995). Selective harvesting robot of cabbage. *Proceedings of ARBIP95* vol. 2, 25–32. Japan Society of Agricultural Machinery.
- Ochs, E. S. & Gunkel, W. W. (1993). Robotic grape pruner field performance simulation. Paper no. 933528. St. Joseph, MI: ASAE.
- Okamoto, T., Kitani O. & Torii, T. (1992). Tissue proliferation robot in plant tissue culture. Paper no. 923516. St. Joseph, MI: ASAE.

- Okamoto, T., Shirai, Y., Fujiura, T. & Kondo, N. (1992). *Intelligent Robotics*. Japan, Tokyo: Jikkyo Shuppan.
- Reed, J. N., He, W. & Tillett, R. D. (1995). Picking mushrooms by robots. *Proceedings of ARBIP95* vol. 1, 27–34. Japan Society of Agricultural Machinery.
- Sevila, F. (1985). A robot to prune the grapevine. *Agri-Mation I*, 190–199. St. Joseph, MI: ASAE.
- Simonton, W. (1990). Automatic geranium stock processing in a robotic workcell. *Transactions of the ASAE* 33(6): 2074–2080.
- Slaughter, D. C. & Harrell, R. C. (1987). Color vision in robotic fruit harvesting. *Transactions of the ASAE* 30(4): 1144–1148.
- Suzuki, M., Onoda, A., & Kobayashi, K. (1993). Development of the grafting robot for cucumber seedlings. *Proceedings of the International Conference for Agricultural Machinery & Process Engineering*, 859–866. Korea: Seoul.
- Tai, Y. W., Ling, P. P. & Ting, K. C. (1994). Machine vision assisted seedling transplanting. *Transactions of the ASAE* 37(2): 661–667.
- Ting, K. C., Giacomelli, G. A. & Shen, S. J. (1990). Robot workcell for transplanting of seedlings part I-layout and materials flow. *Transactions of the ASAE* 33(3): 1005–1010.
- Ting, K. C., Giacomelli, G. A., Shen, S. J. & Kabala, W. P. (1990). Robot workcell for transplanting of seedlings part II-end-effector development. *Transactions of the ASAE* 33(3): 1013–1017.
- Ting, K. C. & Giacomelli, G. A. (1992). Automation-culture-environment based systems analysis of transplant production. *Transplant Production Systems*, 83–102. The Netherlands: Kluwer Academic Publishers.
- Tokuda, M., Namikawa, K., Suguri, M., Umeda, M. & Iida, M. (1995). Development of watermelon harvesting robot (I)-machine vision system for watermelon harvesting robot. *Proceedings of ARBIP95* vol. 2, 9–16. Japan Society of Agricultural Machinery.
- Vassura, G. (1991). Fruit-swallowing oesophagus for a peach-picker robot arm: a feasibility study. *The Second Workshop on Robotics in Agriculture & the Food Industry*, 79–91. Italy: DIST University of Genova.
- Yamada, H., Buno, S., Koga, H., Uchida, K., Ueyama, M., Anbe, Y. & Mori, H. (1985). Development of a grafting robot. *Proceedings of ARBIP95* vol.3, 71–78. Japan Society of Agricultural Machinery.
- Yamashita, J., Satou, K., Fujiura, T., Kondo, N. & Imoto, T. (1992). Agricultural robots (4)-automatic guided vehicle for greenhouses. Paper No. 923544. St. Joseph, MI: ASAE.
- Yang, Y., Ting, K. C. & Giacomelli, G. A. (1991). Factors affecting performance of sliding-needles gripper during robotic transplanting of seedlings. *Applied Engineering in Agriculture* 7(4): 493–498.
- Yoshikawa, T. (1983). Analysis and control of robot manipulators with redundancy. Preprints of the 1st International Symposium of Robotics Research, August 28–September 2.

Three-Dimensional Image Reconstruction Procedure for Food Microstructure Evaluation

K. DING¹ and S. GUNASEKARAN²

¹*Universal Instruments Corporation, Binghamton, NY, USA;*

²*Biological Systems Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA*

Abstract. Confocal laser scanning microscopy (CLSM) is a noninvasive technique for evaluating the microstructure of foods and other materials. CLSM provides several sequential subsurface layers of two-dimensional (2-D) images. An image processing algorithm was developed to reconstruct these 2-D layers into a three-dimensional (3-D) network. Microstructure of fat globules in cheese was used as an example application. The validity of the image reconstruction algorithm was evaluated by processing several layered digital images of known shape and size. Differences between the original and reconstructed images were 2–5% in terms of object size and 1–8% in terms of shape.

Key words: Confocal laser scanning microscopy (CLSM), food microstructure, image processing, computer vision, cheese, fat globules

Introduction

The study of physical and functional properties of foods is hindered by the inherent inhomogeneities of foods. Instrumented measurement of textural properties involves measuring stress, strain, and time effects. The results obtained are in response to forces acting on an underlying organized structure. Perhaps the best approach is to study how sensory and mechanical responses relate to structural organization (Stanley 1987). Observing microstructure and changes in it with perturbations of composition or physical forces can reveal parameters directly related to texture. In fact, it is the microstructure which actually determines the sensory and mechanical characteristics of a food. Foods having similar microstructures can be loosely grouped together as foods that have similar textures (Kalab et al. 1995). Thus, knowledge of microstructure must be antecedent to any plan aimed at either manipulation or regulation of texture.

A variety of microscopic techniques are available to examine food microstructure. These include light microscopy, scanning electron microscopy (SEM), transmission electron microscopy (TEM) and confocal laser

scanning microscopy (CLSM). SEM and TEM offer the advantage of high resolution, but sample preparation procedures are laborious and may lead to artifacts (Kaláb 1984; Liboff 1988). CLSM offers an alternative method to observe food structure without disturbing the internal structure (Brakenhoff et al. 1988; Brooker 1991).

One of the major problems associated with traditional microscopic methods of studying complicated structures is quantification of image features. Computer imaging is an extremely powerful technique for extracting and quantifying features for quality assessment and control. It can also help in understanding mechanisms of complex processes that alter product characteristics. It offers the advantages of accurate quantification of images and rapid data handling. Almost all instruments used to provide an enlarged view of food systems can be used with image analysis, either through direct interfacing to a light or electron microscope or by scanning outputs such as photographs or negatives (Stanley 1987).

Quantitative evaluation of food microstructure is increasing. Ruegg and Moor (1987) determined the size distribution and shape of curd granules in hard and semi-hard Swiss cheeses. The quantification of microstructure is highly facilitated by computer image analysis (Inoue 1986). Within the past few years, computer image processing has become increasingly popular in the food industry (Gunasekaran and Ding 1994; Gunasekaran 1996). Applications range from quality parameters of food grains (Gunasekaran et al. 1987; Gunasekaran et al. 1988) and microscopic characteristics of leaves (Wen and Gunasekaran 1992) to intramuscular fat in muscle (Ishii et al. 1992). Ali and Robinson (1985) used a digitizer tablet to transfer micrograph images for analysis to evaluate size distribution of casein micelle in camel milk. Recently, Holcomb et al. (1992) investigated textural changes in dairy products by image analysis. Ding and Gunasekaran (1992 and 1993) presented preliminary results of microstructure study of cheese and milk gel using computer image processing. An example of the use of digital image analysis to quantify structural features of cheese was given for string cheese (Taneya et al. 1992).

We recently applied confocal microscopy to compare the size and shape of fat globules in reduced-fat Cheddar cheese where the globules were coated either with native milk proteins in a non-specific fashion, lactalbumin (heat-denatured β -lactoglobulin and α -lactoalbumin) or β -casein (Everett et al. 1995). These results were based on analysis of 2-D layered images. In order to describe the internal structure more completely, a 3-D network would be most suitable. Owing to the development CLSM and the increased computational power and speed of computers, 3-D microscopy is becoming the latest trend in the microstructural analysis of foods (Gunasekaran 1996; Vodovotz

et al. 1996; Blonk and van Aalst 1993). There are some commercial 3-D reconstruction softwares suitable for CLSM images. However, they are either too expensive or too specific for biomedical-type applications.

Our objectives were: 1) to develop a 3-D image reconstruction algorithm from 2-D layered CLSM images and 2) to validate the algorithm by processing several known 3-D elements.

Confocal Laser Scanning Microscopy

Confocal microscopy is characterized by the ability to make selective observations of one plane within a specimen, unaffected by the image of out-of-focus regions above and below the plane. This technique is known as optical slicing (Rao et al. 1992). In Figure 1, the optical slicing technique used in the CLSM is illustrated schematically. When a conventionally imaged object is displaced from best focus, the image contrast decreases but the spatially averaged intensity remains the same. In a confocal system, however, the image of defocused surface appears darker than if it were in focus. Thus, confocal optics can be said to have axial resolution in addition to lateral resolution. As a consequence, it is possible to extract topographic information from a set of confocal images taken over a range of focal planes. In order to form a confocal image, the signal is recorded as the object is scanned relative to the image of the point source in a plane parallel to the focal plane. Multiple confocal image slices are obtained by repeating the process at various levels of object defocus. By focusing at different heights (along the z-axis) on the object, a 3-D topographical map of the object is obtained. The resolution in the z-direction (axial resolution, Δz) depends upon the numerical aperture of the lens, the degree to which the pinhole is open, and the wavelength of the laser light. If the confocal pinhole is fully opened, the microscope becomes a conventional scanning light microscope with reduced lateral resolution and larger depth of field in the z-direction. The minimum Δz is $0.5 \mu\text{m}$ (Brakenhoff et al. 1988) which is very difficult to achieve using a regular light microscope. Maximum observation depths of $10 \mu\text{m}$ to $100 \mu\text{m}$ can be achieved, depending on the opacity and absorption characteristics of the specimen (Heertje et al. 1987; Brakenhoff et al. 1988; Brooker 1991). In addition, specimen observations can be made within a plane both transverse to and along the optical axis, compared to conventional light microscopy where only images transverse to the optical axis can be made (Brakenhoff et al. 1988). Perhaps a disadvantage of the CLSM is that the minimum resolvable area is approximately $0.2 \mu\text{m} \times 0.2 \mu\text{m}$. For additional description of the CLSM the readers are referred to a recent article by Vodovotz et al. (1996).

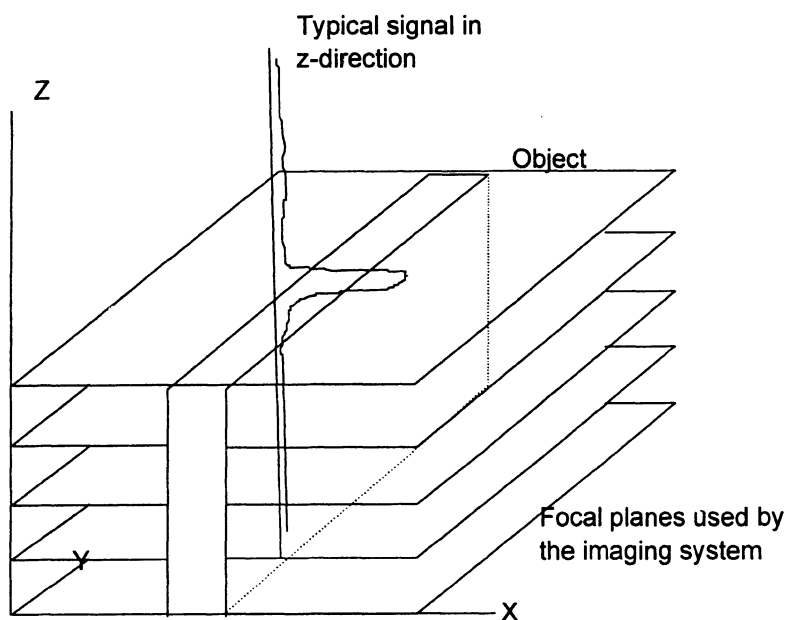


Figure 1. Schematic representation of optical slicing in CLSM (based on Rao et al. 1992).

Image Reconstruction

Since images obtained from a microscope are specific to the settings used, the following was considered in developing the 3-D reconstruction algorithm. Microstructure of fat globules in cheese is used as the example application. However, the procedure described here and the algorithm developed is suitable for evaluating similar images. Confocal images were taken using an MRC-600 confocal microscope (Bio-Rad Microscience, Ltd.) available at the Integrated Microscopy Resource at the University of Wisconsin-Madison. Additional details of the imaging process used (staining, lens etc.) are described in Ding (1994). The resolvable area called "pixel" is an element square of a 2-D image that was approximately $0.3\ \mu\text{m} \times 0.3\ \mu\text{m}$. The separation between observation planes was set at $0.5\ \mu\text{m}$, which is the smallest separation for the $40\times$ magnification lens. For each specimen, 81 adjacent planes (layers) were observed. Thus, the observation depth was $40\ \mu\text{m}$. The 3-D image was 256×256 pixels and 8-bit gray-scale image (256 levels of gray). CLSM provides a sequence of 2-D layered images. Some 2-D image layers of the cheese are presented in Figure 2. The 3-D image of the fat globules in Cheddar cheese reconstructed per the procedure described herein is shown in Figure 3.

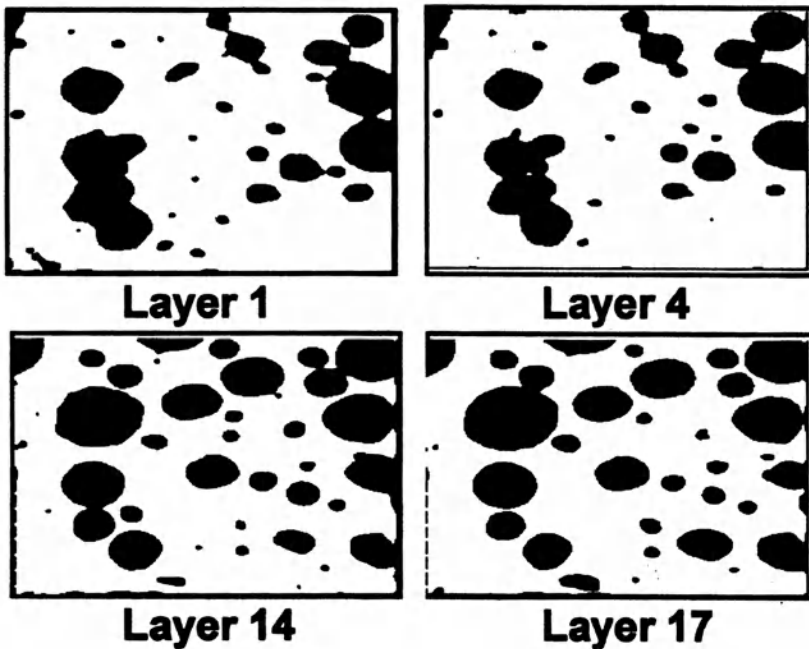


Figure 2. Some 2-D images from CLSM showing fat globules in Cheddar cheese (black areas) at different subsurface layers.

Before the layered images were reconstructed, they were subjected to background correction and scaled as explained below. The image analysis software Optimas (Bioscan Corp., Seattle, WA) was used as the base for developing algorithms.

Background correction

Because of the lens and lighting problems, 2-D CLSM micrographs often are brighter in the center and fade out gradually towards the edges. An uneven background may make it impossible to set a single gray scale threshold value for image segmentation.

To even out the background, a quadratic polynomial surface was used to fit the background by statistical regression. Then the quadratic polynomial surface was subtracted from the original image surface. The average intensity of the image was added back to this. To simplify the description, a line of a 2-D image is selected as an example, and the background correction procedure is illustrated in Figure 4a. The original line of the image has low intensity in the middle and high intensity on either side. The corrected line of the image has fairly uniform intensity throughout. Figure 4b shows a 2-D layered image of

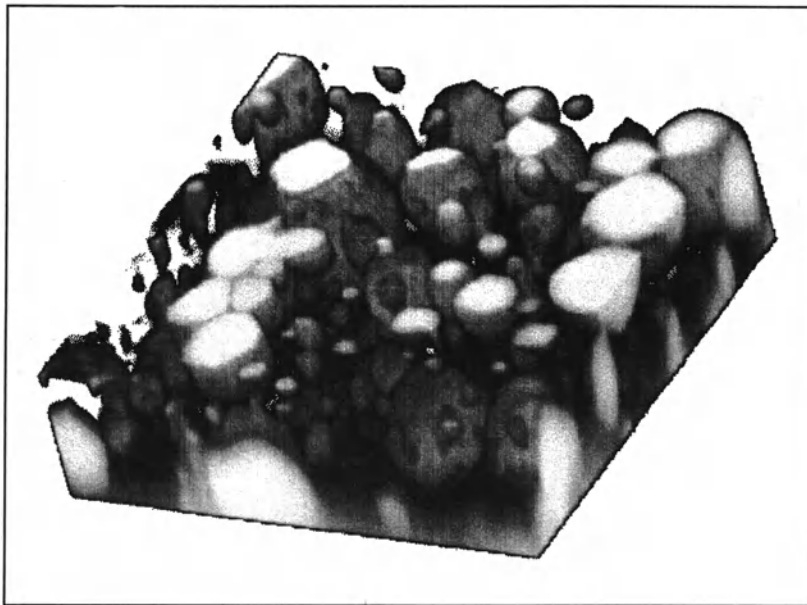


Figure 3. A 3-D image of fat globules in Cheddar cheese reconstructed from 2-D image layers from CLSM (the brighter areas represent fat globules chopped during specimen preparation).

the cheese *before* the background correction. (The central part is darker than other parts because this is an intensity reversed image, i.e., black and white are reversed for image processing, of a CLSM micrograph.) Figure 4c shows the same 2-D image layer *after* the background correction. This image has an approximately even background.

The background can only be corrected approximately, however, and some minor information about the image may be lost during the process. Before background correction, some small fat globules at the image boundary are visible. After the background correction, they are invisible. Since the lost information is usually small, the background correction technique has been widely used for microscopic imaging.

Image segmentation

Image segmentation is normally achieved by setting a threshold based on image histogram. However, because of the lighting problem, the brightness in different layers of the CLSM slices was different. Thus, even after background correction, although it was possible to use a single threshold for one 2-D layer, it was still impossible to set a fixed threshold for all layers. In addition, because it was very hard to obtain proper bimodal histograms of images from

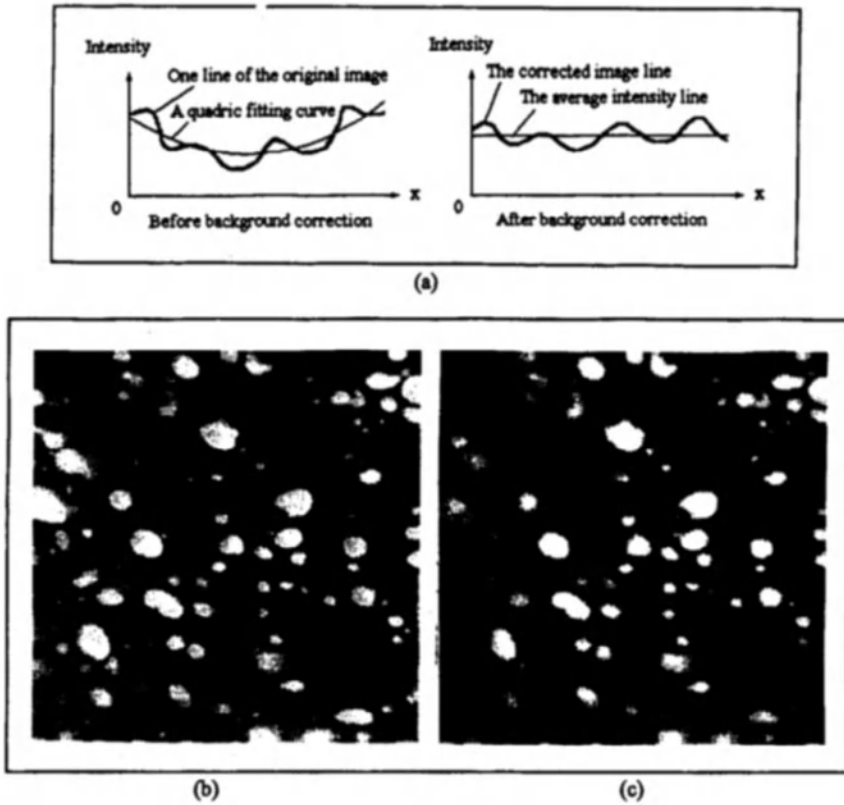


Figure 4. Description of background correction procedure (a); and an image before (b) and after (c) background correction.

all layers, the optimal thresholding cannot properly segment fat globules from the background. Since there was no shrinkage during the CLSM sample preparation, the threshold of each 2-D layer could be set according to the overall fat content of cheese assuming that the fat was distributed uniformly. Given the small sample size used, this is a very reasonable assumption. The fat content of the low-fat Cheddar cheese used in this study was 13.9%. Thus, 13.9% of the brighter pixels in each 2-D layer was segmented as fat globules (i.e. set as white pixels). For thresholding and other processing operations we used the Optimas Image Analysis Software (Optimas Corporation, Bothell, WA) as the base program.

Image scaling

The width of a 2-D image layer was 77 μm . There were 256×256 pixels per image, so the size of one pixel is $0.3 \mu\text{m} \times 0.3 \mu\text{m}$. A 2-D image is processed pixel-by-pixel. A 3-D image is processed voxel-by-voxel. A voxel is an element of a 3-D image that should be a cube (Figure 3) in order to be easily processed. However, the distance between two adjacent layers was 0.5 μm which did not equal the pixel size ($0.3 \mu\text{m} \times 0.3 \mu\text{m}$) in a 2-D image layer. Therefore, the voxel shape was not cubical. To reconstruct the 3-D image that has cubic-shaped voxels, image scaling was performed using the built-in function of the Optimas software. After scaling all 2-D layered images, they could be put together to construct a 3-D image. Since the 2-D images were digitized, each could be represented as a 2-D matrix. Thus, a 3-D image was a 3-D matrix and all the voxel dimensions were $0.5 \mu\text{m} \times 0.5 \mu\text{m}$.

Object identification

The image layers were scanned from top to bottom and left to right. The microstructural elements of interest (fat globules) were numbered voxel-by-voxel by a process known as image labeling. The digitized voxel values (intensities) were the elements of the 3-D matrix. Figure 5 illustrates the schematic diagram of a $6 \times 6 \times 6$ 3-D matrix. Before scanning, the 3-D image was a binary image (a binarized 3-D matrix).

White voxels were labeled as 0 and black voxels were labeled by a large number b . We selected b to be larger than the maximum expected number of globules (or objects of interest). We set $b = 32768$ because it represents half the possible number of pixels (i.e., $(256 \times 256)/2$).

During scanning, the boundary voxel values were checked in order to distinguish the boundary-chopped and non-chopped fat globules, and the rest of the 3-D image was scanned. For the entire 3-D image, the scanning was actually from the second top layer to the second bottom layer (from layer 1 to layer 4 in Figure 5). For each layer, the scanning was from the second front row to the second back row (from row 1 to row 4 in Figure 5); for each row, the scanning was from second left voxel to the second right voxel (from column 1 to column 4 in Figure 6). To label the boundary-chopped fat globules and boundary non-chopped fat globules separately, two stacks of numbers were selected. Stack C (1 to 1000 with 1 on the top of the stack) was for the boundary-chopped fat globules (reflects our assumption that there were fewer than 1000 boundary-chopped fat globules in the specimen being examined). Stack N (1001 to 32,767 with 1001 on the top of the stack) was

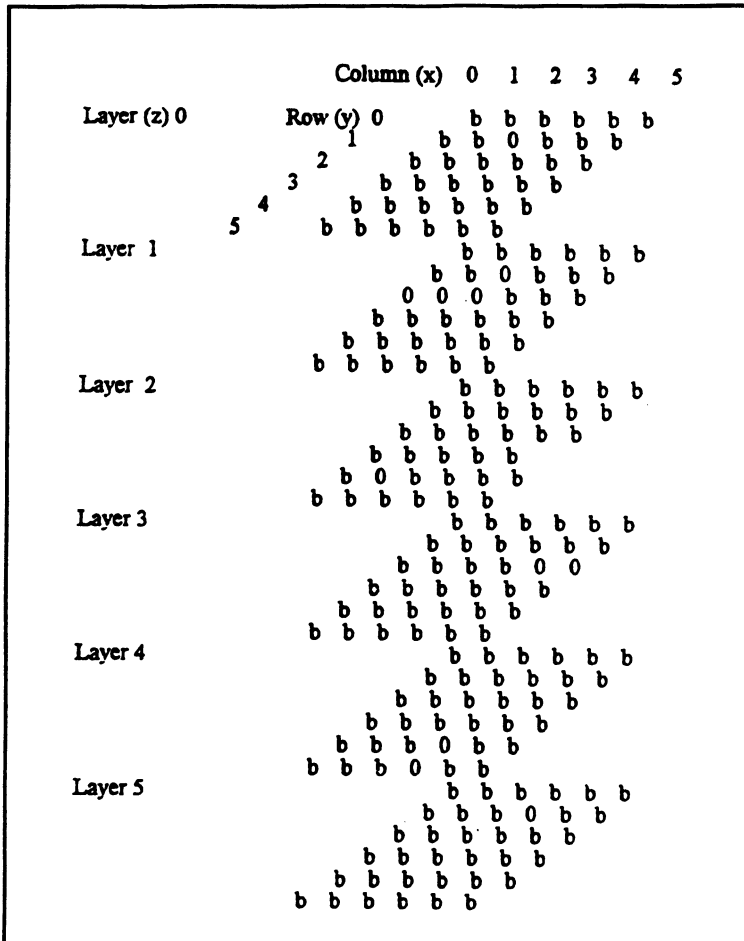


Figure 5. Schematic diagram of a 3-D matrix of an image before numbering.

for the boundary non-chopped fat globules. During scanning, the algorithm was as follows.

- (1) For each line scanned, only the white voxels (elements of 0 in Figure 5) were processed as they belong to fat globules (or the object of interest) in the 3-D image. It saved much processing time, especially for very low-fat cheeses.
- (2) For each white voxel (voxel value = 0), the voxel values of the top, front and left adjacent voxels were checked. If the smallest voxel value of the three adjacent voxels was zero, that voxel may be part of a fat globule that

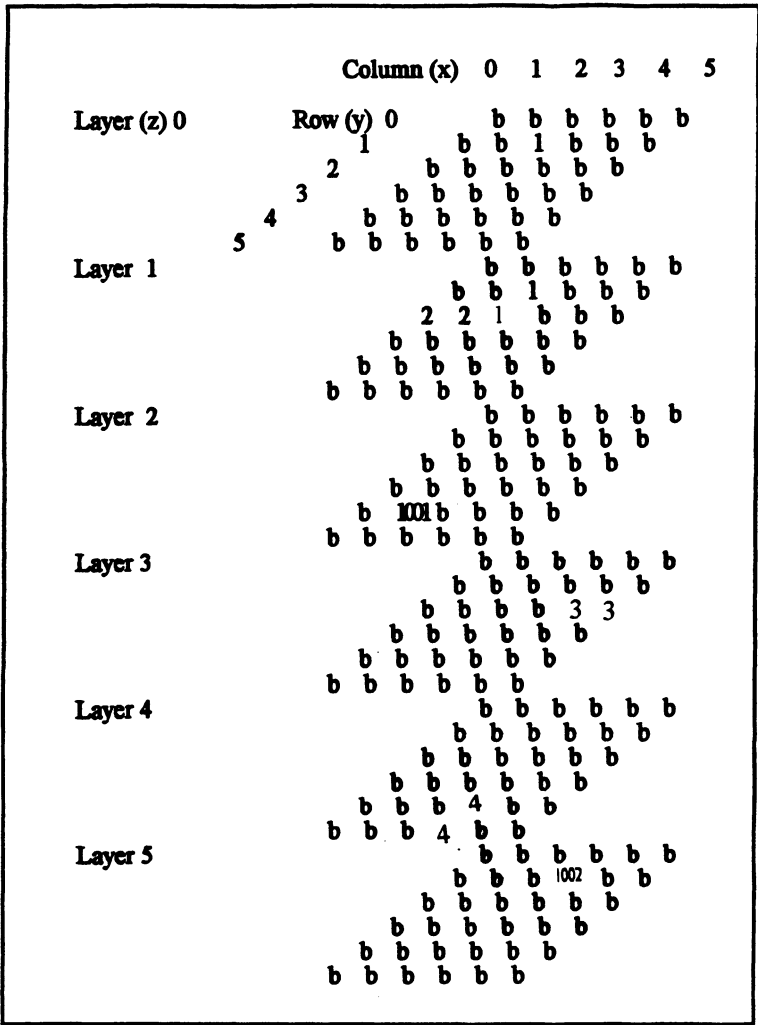


Figure 6. Schematic diagram of a 3-D matrix of an image during numbering.

has not been labeled. The algorithm first checks whether the fat globule was chopped by the boundary of the 3-D image as follows.

If one of the three adjacent voxels was a boundary white voxel, that indicates that at least one voxel of the fat globule was on the boundary of the 3-D image. So that fat globule was recognized as a boundary-chopped fat globule. One number on the top of stack C was used. The voxel and the boundary voxel were labeled by that number. For example, the voxel

- $[x = 2, y = 1, z = 1]$ should be set to 1 because the top adjacent voxel $[x = 2, y = 1, z = 0]$ was a boundary white voxel. The voxel $[x = 1, y = 2, z = 1]$ should be set to 2 because the adjacent voxel on the left $[x = 0, y = 2, z = 1]$ was a boundary white voxel. Otherwise, the fat globule was not boundary-chopped. One number on the top of stack N was used, and the voxel was labeled by that number. For example, the voxel $[x = 1, y = 4, z = 2]$ should be set to 1001 (the first boundary non-chopped fat globule).
- (3) If the smallest value of the three adjacent voxels was greater than zero, that indicates that the fat globule had been labeled and the voxel belonged to the labeled fat globule. Therefore, the voxel was labeled by the smallest number. For example, in Figure 5 the voxel $[x = 2, y = 2, z = 1]$ should be set to 1 because the smallest value of the adjacent voxel $[x = 2, y = 1, z = 1]$ was 1.
 - (4) If the second or third smallest number was less than the background voxel number ($< 32,767$ in our program), that indicates that the fat globule had been labeled by two or more numbers in the upper layers or front lines. In Figure 6, the fat globule of voxels $[x = 2, y = 1, z = 1]$, $[x = 2, y = 2, z = 1]$ and $[x = 1, y = 2, z = 1]$ were labeled by two numbers (1 and 2). Whenever this case occurred, the algorithm would track back and relabel all the voxels to be the same number.

Smaller numbers were always used for relabelling. The second or third smallest number was returned to stack C or stack N, depending on where the number was from. In the case immediately above, the voxels of the fat globule would be relabelled 1; number 2 would be returned back to stack C. So, under normal processing, the voxels labeled 3 and 4 in Figure 6, would have been labeled 2 and 3, respectively. Returning the numbers and reusing them off the top of the stack helped to save computer memory. If all the fat globules in a 3-D image can be labeled by 255 numbers, the data type byte (8 bits) can be used instead of an integer (32 bits) to save memory. A listing of the computer program implementing this algorithm is in Ding (1995).

Algorithm Validation

Since the exact size and shape of the fat globules in the cheese is not known we have to select a known standard for validating our algorithm. Therefore, following objects were constructed in 23 layers of 2-D digital images (2-D matrixes, Figure 7). The objects were: two spheres (diameter $d = 5 \mu\text{m}$ and $10 \mu\text{m}$, respectively); one vertical cylinder (diameter $d = 5 \mu\text{m}$, height $h = 5.5 \mu\text{m}$); one horizontal cylinder (diameter $d = 5 \mu\text{m}$, height $h = 6.0 \mu\text{m}$); one cylinder (diameter $d = 5 \mu\text{m}$, height $h = 5.0 \mu\text{m}$) slanted 59 to horizontal plane; and one vertical cylinder (diameter $d = 10 \mu\text{m}$, height $h = 10 \mu\text{m}$) with

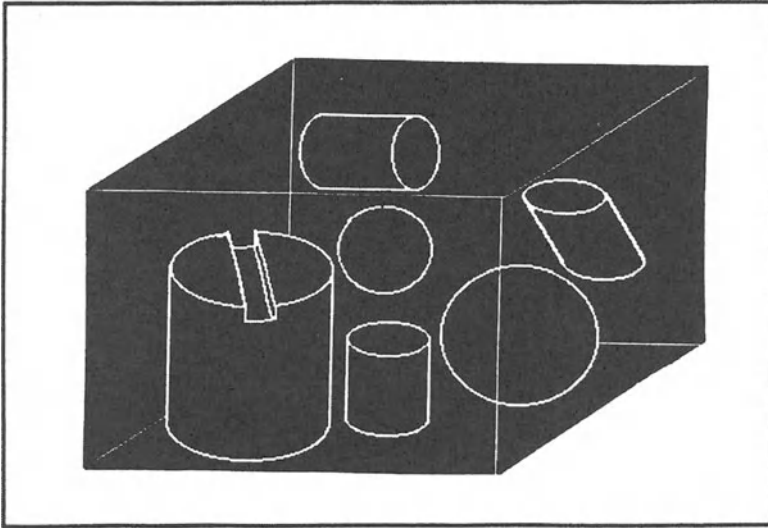


Figure 7. Schematic of different objects constructed for algorithm validation.

a notch (width $w = 3 \mu\text{m}$, length $l = 10 \mu\text{m}$, height $h = 1.5 \mu\text{m}$) on the top. The procedure used to construct these images is as follows.

2-D layered image construction

A 2-D layered image is a 2-D matrix of pixels. The objects were assigned a pixel value of 255 (white) and the background 0 (black). Since pixel size and distance between adjacent layers of the CLSM layered images in our experiments were known (pixel size = $0.3 \mu\text{m} \times 0.3 \mu\text{m}$, distance between adjacent layers = $0.5 \mu\text{m}$), the 2-D layered validation images were constructed exactly the same as the 2-D layered CLSM images. The coordinate system used is shown in Figure 8 where each pixel location in the image of layer i was determined by:

$$X_i = \text{Row number} * \text{Pixel width } (0.3 \mu\text{m});$$

$$Y_i = \text{Column number} * \text{Pixel length } (0.3 \mu\text{m});$$

$$Z_i = \text{Layer number} * \text{Distance between two layers } (0.5 \mu\text{m}).$$

To construct a sphere, the process was initiated at the center of the sphere $[X_c, Y_c, Z_c]$ (Figure 8). The distance between each pixel location $[X_i, Y_i, Z_i]$ and the center of the sphere $[X_c, Y_c, Z_c]$ was calculated. If the distance was greater than the sphere radius R , the pixel was assigned 0 (black). Otherwise the pixel was assigned 255 (white).

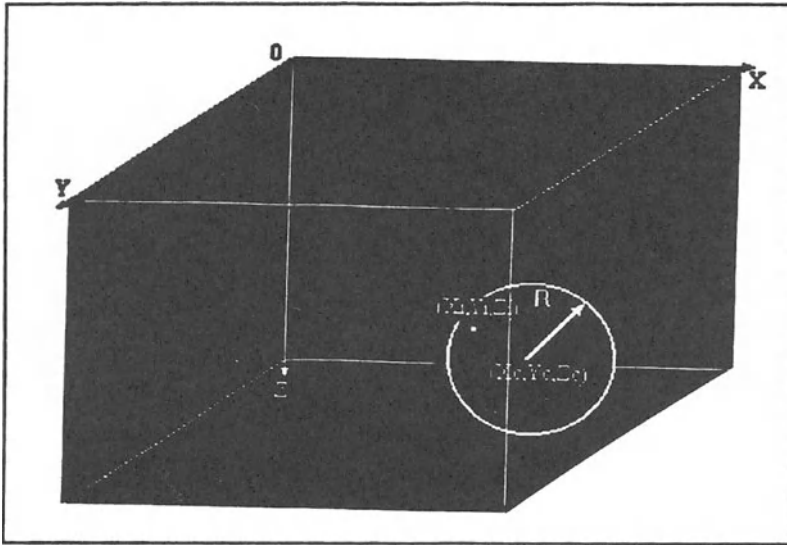
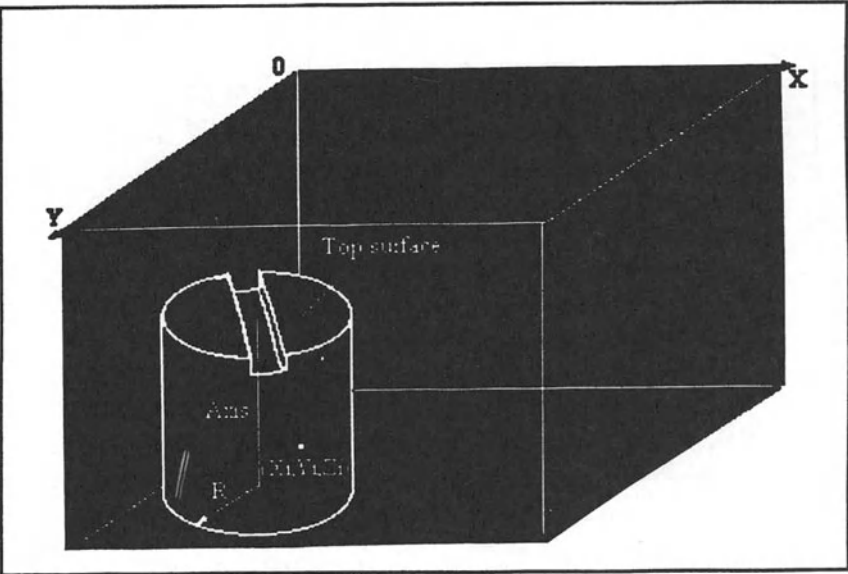


Figure 8. Details of construction of a sphere.

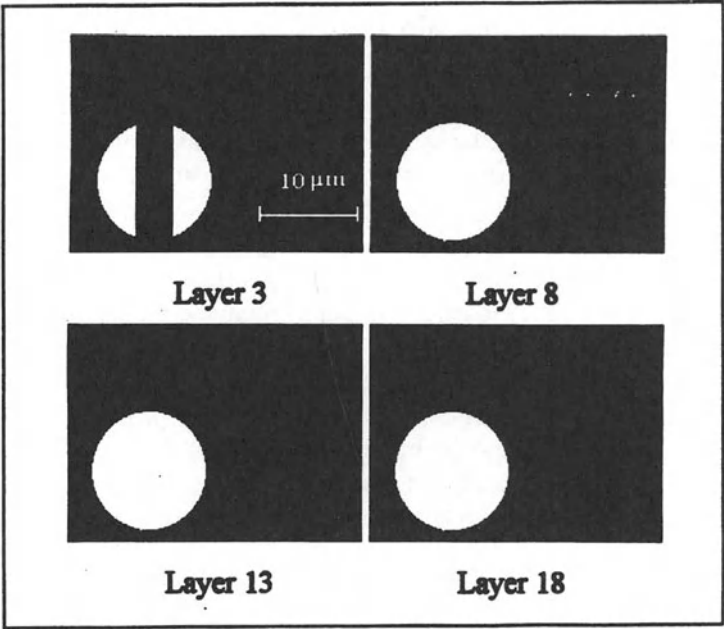
To construct a cylinder, the top surface and axis of the cylinder were the starting points (Figure 9). The distance between each pixel $[X_i, Y_i, Z_i]$, and the axis of the cylinder was calculated. In addition, the distance between each pixel $[X_i, Y_i, Z_i]$ to the top surface was calculated. If the distance indicated that the pixel is outside of the cylinder, the pixel was assigned 0; otherwise it is assigned 255. Similarly, after the axis and the width of the notch were known, the pixel $[X_i, Y_i, Z_i]$ was easily found in the cylinder with or without a notch. Similarly, all the objects shown in Figure 7 were constructed in 2-D layers. These 2-D layered images were reconstructed using the algorithm developed. To verify the closeness of the original and reconstructed images, the ideal size and number of surface voxels obtained by geometrical models were compared with the calculated size (diameter) and surface voxel number of the reconstructed image obtained by the 3-D image processing algorithm. For a given geometric shape the number of surface voxels can be calculated (Figure 10) and thus can be used for validating the object shape. The results are shown in Table 1. The relative difference of sphere diameter is about 2–4%, and the relative difference of surface voxel number is about 1–8%. The relative difference was defined as:

$$\text{Relative difference} = \frac{|\text{Actual value} - \text{Reconstructed value}|}{\text{Actual value}}$$

All objects in the reconstructed 3-D image were successfully separated by the 3-D image processing algorithm and numbered differently. The boundary-



(a) Overall shape of a cylinder with a notch.



(b) Binary images of four 2-D layers of the above.

Figure 9. Details of construction of a cylinder with a notch.

Table 1. Validation of 3-D image reconstruction algorithm by comparing size and number of surface voxels of several objects of known size and shape

Object	Diameter (μm)		Relative difference ¹	No. of surface voxels		Relative difference ¹
	Actual	Reconstructed		Actual	Reconstructed	
Sphere	5.0	5.2	0.04	256	236	0.08
Sphere	10.0	10.2	0.02	1028	1011	0.02
Vertical cylinder	5.9	6.1	0.03	412	404	0.02
Horizontal cylinder	6.2	6.5	0.04	468	494	0.06
Inclined cylinder	5.7	5.9	0.02	382	380	0.01
Cylinder with notch	11.4	11.3	0.02	1718	1621	0.06

¹ Relative difference = (Actual – Reconstructed)/Actual.

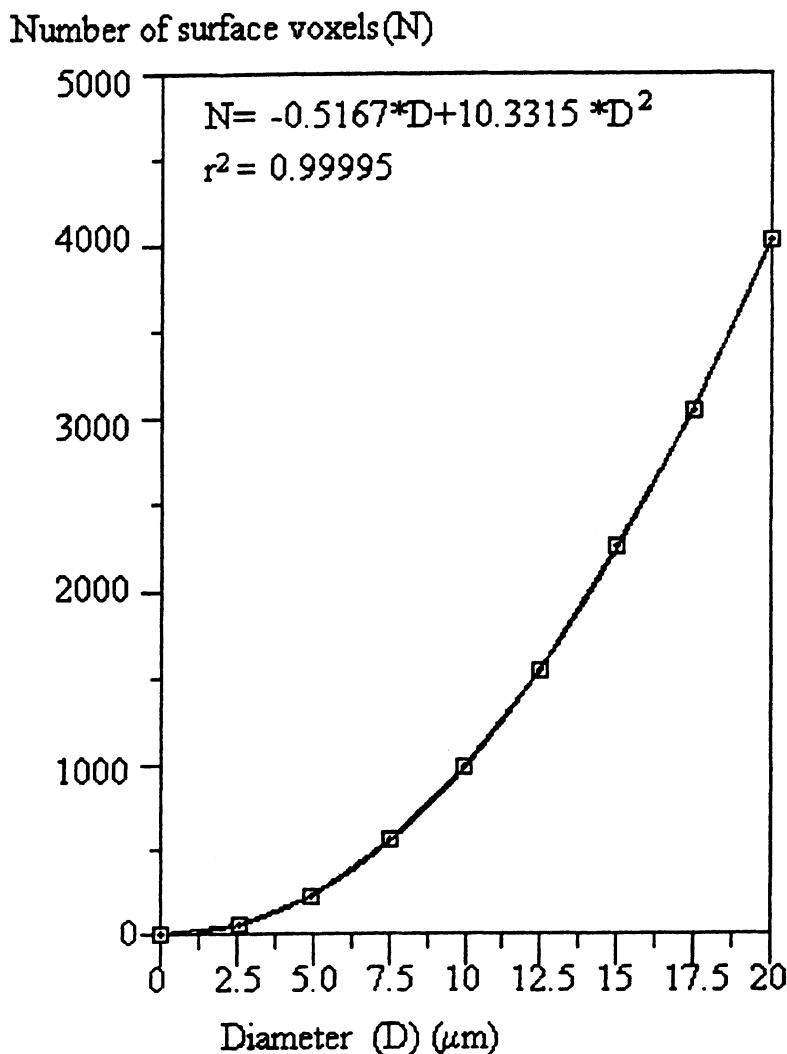


Figure 10. Number of surface voxels vs. diameter of a sphere.

touching object, the larger sphere, was also recognized by the 3-D image processing algorithm correctly. The concave-shaped object, the cylinder with notch, was initially numbered as two objects in the top layers. The 3-D image processing algorithm successfully recognized them to be parts of the same object when it processed the bottom layer of the notch and backtracked correctly.

Conclusions

An image-processing algorithm was developed to reconstruct several sequential 2-D image layers obtained from a CLSM into a 3-D image. An example application of reconstructing fat globules in cheese was considered. The algorithm was evaluated for its validity by reconstructing several 2-D layered digital images of objects of known size and shape. The algorithm performed very satisfactorily. Differences between the original and reconstructed images were 2–5% in terms of object size and 1–8% in terms of shape.

References

- Ali, M. Z. & Robinson, R. K. (1988). Size Distribution of Casein Micelles in Camel's Milk. *J. Dairy Res.* **52**: 303–307.
- Blonk, J. C. G. & van Aalst, H. (1993). Confocal Scanning Light Microscopy in Food Research. *Food Res. Int.* **26**(4): 297–311.
- Brakenhoff, G. J., van der Voort, J. T. M., van Spronsen, E. A. & Nanninga, N. (1988). 3-Dimensional Imaging of Biological Structures by High Resolution Confocal Scanning Laser Microscopy. *Scanning Microscopy* **2**(1): 33–40.
- Brooker, B. E. (1991). The Study of Food Systems Using Confocal Laser Scanning Microscopy. *Microscopy and Analysis* (Nov.): 13–15.
- Ding, K. & Gunasekaran, S. (1992). Image Processing for Cheese Microstructure Characterization. *Presented at Food Structure 1992 Meeting of Scanning Electron Microscopy International*. Chicago, IL.
- Ding, K. & Gunasekaran, S. (1993). Milk Gel and Microstructure Classification Using Machine Vision. *Presented at 1993 IFT Meeting*. Chicago, IL.
- Ding, K. (1995). *Food Shape and Microstructure Evaluation with Computer Vision*. Unpublished Ph.D. Thesis, University of Wisconsin-Madison, Madison, WI.
- Everett, D., Ding, K., Olson, N. & Gunasekaran, S. (1995). Applications of Confocal Microscopy to Fat Globule Structure in Cheese. In Malin, E. L. & Tunick, M. H. (eds.) *Chemistry of Structure/Function Relationships in Cheese*, 321–330. Plenum Publishing Corp.: New York, NY.
- Gunasekaran, S., Cooper, T. M., Berlage, A. G. & Krishnan, P. (1987). Image Processing for Stress Cracks in Corn Kernels. *Trans. of the ASAE* **30**(1): 266–271.
- Gunasekaran, S., Cooper, T. M. & Berlage, A. G. (1988). Evaluating Quality Factors of Corn and Soybean Using a Computer Vision system. *Trans. of the ASAE* **31**(4): 1264–1271.
- Gunasekaran, S. & Ding, K. (1994). Using Computer Vision for Food Quality Evaluation. *Food Technol.* **48**(6): 151–154.
- Gunasekaran, S. (1996). Computer Vision Technique for Food Quality Assurance. *Trends in FoodSci. and Tech.* **7**(8): 245–256.
- Heertje, I., van der Vlist, P., Blonk, J. C. G., Hendrickx, H. A. C. M. & Brakenhoff, G. J. (1987). Confocal Scanning Laser Microscopy in Food Research: Some Observations. *Food Microstructure* **6**(2): 115–120.
- Holcomb, D. N., Pechak, D. G., Chakrabarti, S. & Opsahl, A. (1992). Visualizing Textural Changes in Dairy Products by Image Analysis. *Food Technology* **44**(1): 122.
- Inoue, S. (1986). *Video Microscopy*. Plenum Press: New York, NY.
- Inoue, S. (1995). Foundations of Confocal Scanned Imaging in Light Microscopy. In J. B. Pawley (ed.) *Handbook of Biological Confocal Microscopy*, 1–14. Plenum: New York.
- Ishii, T., Cassens, R. G., Scheller, K. K., Arp, S. C. & Schaefer, D. M. (1992). Image Analysis to Determine Intramuscular Fat in Muscle. *Food Structure* **11**: 55–60.

- Kalab, M. (1984). Artifacts in Conventional Scanning Electron Microscopy of Some Milk Products. *Food Microstructure* 3(2): 95–111.
- Kalab, M., Allan-Wojtas, P. & Miller, S. S. (1995). Microscopy and Other Imaging Technique in Food Structure Analysis. *Trends in Food Sci. and Tech.* 6(6): 177–186.
- Liboff, M., Goff, H. D., Haque, Z., Jordan, W. K. & Kinsella, J. E. (1988). Changes in the Ultrastructure of Emulsions as a Result of Electron Microscopy Preparation Procedures. *Food Microstructure* 7(1): 67–74.
- Rao, A. R., Ramesh, N., Wu, F. Y., Mandville, J. R. & Kerstens, P. J. M. (1992). Algorithms for a Fast Confocal Optical Inspection System. *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 298–305.
- Rosenberg, M., McCarthy, M. J. & Kauten, R. (1991). Magnetic Resonance Imaging of Cheese Structure. *Food Structure* 10: 185–192.
- Ruegg, M. & Moor, V. (1987). The Size Distribution and Shape of Curd Granules in Traditional Swiss Hard and Semi-Hard Cheeses. *Food Microstructure* 6: 35–46.
- Stanley, D. W. (1987). Food Texture and Microstructure. In H. R. Moskowitz (ed.) *Food Texture*. Marcel Dekker, Inc.: New York, NY.
- Taneya, S., Izutsu, T., Kimura, T. & Shioya, T. (1992). Structure and Rheology of String Cheese. *Food Structure* 11: 61–71.
- Velinov, P. D., Cassens, R. G., Greaser, M. L. & Fritz, J. D. (1990). Confocal Scanning Optical Microscopy of Meat Products. *J. of Food Sci.* 55(6): 1751–1752.
- Vodovotz, Y., Vittadini, E., Coupland, J., McClements, D. J. & Chinachoti, P. (1996). Bridging the Gap: Use of Confocal Microscopy in Food Research. *Food Technology* 50(6): 74–82.
- Wen, Y. & Gunasekaran, S. (1992). *Measuring Stomatal Characteristics of Elm Leaves by Machine Vision System*. ASAE Paper No. 927022. ASAE Int. Summer Meeting, Charlotte, NC.
- Whallon, J. H., Preller, A. & Wilson, J. E. (1994). Reflection Confocal Imaging of Type I and type III Isozymes of Hexokinase in PC12 Cells. *Scanning* 16(2): 111–117.